TAMPERE UNIVERSITY OF TECHNOLOGY

**RIGOBERTO CABRERA**

# APPLICATION OF AN INDUSTRIAL ROBOT IN MASTER-SLAVE TELEOPERATION WITH HAPTIC INTERFACE

Master of Science Thesis

Examiners: Professor Matti Vilenius and
MSc. Ali Muhammad

Examiner and topic approved in the Faculty of Automation, Mechanical and Materials Engineering Council meeting on 9th September 2009

# ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Machine Automation
**CABRERA, RIGOBERTO**: Application of an industrial robot in master-slave teleoperation with haptic interface.
Master of Science Thesis, 52 pages, 2 Appendix pages
May 2010
Major subject: Hydraulic Vehicle Engineering
Examiners: Prof. Matti Vilenius
MSc. Ali Muhammad
Keywords: Robot Teleoperation, Haptics, Kinematics.

The implementation of a master-slave teleoperation system as a technique to perform complex tasks in hazardous environments has been in use for several decades. The results of previous systems developed in the nuclear industry have excellent performance. However, the implementation of every use case has specific requirements that consume a lot of time and resources; the reduction of time in development and budget is an important factor of the system success nowadays. This thesis describes the implementation of a teleoperation system based in the operation of an industrial robot, the already made software, hardware tools and the interaction with a third party software for the system implementation.

The main goal is to implement a conventional teleoperation system with and additional functionalities, which permit to manually manipulate the robot, control the position of the joints and provide a bilateral force-reflection as well. The development is basically divided in two systems. The first system is related with the interaction of the hardware and software that provides the basic teleoperation platform. Such system requires the knowledge of different programming languages, the know-how of mechatronic systems and a high level expertise for the integration of technologies.

The second part of the system focuses on the computation of the parameters related to the dynamics of the robot and the real-time performance of the teleoperation system. The creation of interfaces for monitoring the control loops in the haptic device is the result of the implementation of this thesis work.

The configuration of the teleoperation system on an industrial robot platform is the main limitation; the system has a closed-architecture, where programming or change main variables is not allowed. The implementation is limited by existing software tools and the guidelines provided by the robot manufacturer. However, these tools have the advantage of reducing the implementation time, which has a direct impact on the budget of the project.

Another advantage of this implementation is the utilization of 3D software that provides a state of the arts visual environment, enhancing the user immersion. This gives the operator another tool for reference and facilitates the realization of his tasks.

# PREFACE

The research and project implementation of the present thesis has been carried out during the years 2009-2010 at the Department of Intelligent Hydraulics and Automation (IHA), Tampere University of Technology, Finland. It is part of the efforts to develop a suitable haptic device for the Water Hydraulic Manipulator (WHMAN) of the International Thermonuclear Experimental Reactor (ITER).

I would like to extend my sincere gratitude to my supervisors Prof. Matti Vilenius and MSc. Ali Muhammad for the opportunity to collaborate in this important project, and for their guidance throughout the thesis. Particularly thanks to MSc. Ali Muhammad for his patience and advice during the writing process. I also want to thank Dr. Tech. Jouni Mattila for the financial support provided for the experiment.

Special thank to my parents Rigoberto and Hipolita for their support during this journey and all my life. Sincerely grateful with my fiancée for her understanding, love and encouragement during my studies. Thanks to all the rest of my family and friends.

Tampere, Finland. May 2010

*Rigoberto Cabrera*

# TABLE OF CONTENTS

# TABLE OF FIGURES

# ACRONYMS AND DEFINITIONS

| | |
|---|---|
| **PCIF** | PC Interface Option |
| **PCDK** | PC Developer's Kit |
| **TCP/IP** | Transmission Control Protocol/Internet Protocol |
| **UDP** | User Datagram Protocol |
| **COM** | Component Object Model |
| **RPC** | Remote Procedure Call |
| **ITER** | International Thermonuclear Experimental Reactor |
| **TUT** | Tampere University of Technology |
| **WHMAN** | Water Hydraulic Manipulator |
| **EE** | End-Effector |
| **CoG** | Centre of Gravity |
| **DoF** | Degree of Freedom |
| **DHP** | Denavit Hatemberg Parameters |
| **FK** | Forward Kinematics |
| **IK** | Inverse Kinematics |

# 1. INTRODUCTION

The term *robot* is defined as a machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer [1]. However there is no one universal definition for a robot that satisfy everyone, the variety of appearances and movements difficult the identification of which machine qualify as a robot.

Exist a variety of classifications for robotics, one general classification divides the field in two main areas of research: Advanced robotics and Industrial robotics[2]. Advanced robotics is the discipline studying robots with marked characteristics of autonomy, and Industrial robotics is the discipline concerning robot design, control and applications in industry.

With the necessity to standardize the concept of *Industrial Robot,* the International Organization for Standardization (ISO) define the term as: automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications [3]. The standard divides the entire industrial robot field according to the mechanical structure: Cartesian (Gantry) robots, SCARA robots, articulated robots, parallel robots.

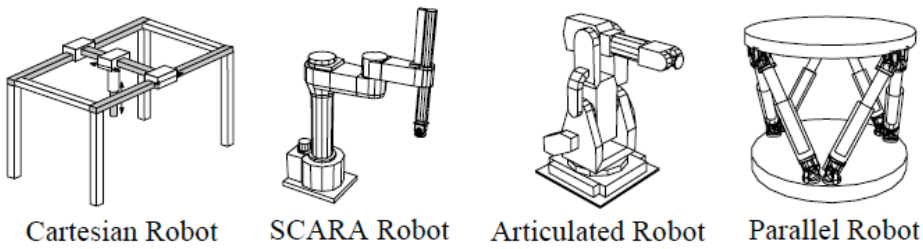



**Figure 1.1** Classification by mechanical structure (from ISO Standard 8373)[3]

Another classification of a robot is related to how the robot is controlled, this classification is continuously evolving according to the demand of new and more complex applications [4]. Based on those requirements, the industrial robot is designed to realize a variety of manufacturing and system processes, where the typical applications include material handling, part transfer, and assembly material removal.

For the way of how the Industrial Robot executes those applications exist another classification in the ISO standard 8373. The classification is according to the following control types: sequence-controlled robot, trajectory operated robot, adaptive robot, and teleoperated robot [3]. All the control types has its pros and cons to accomplish the applications, but the one that is relevant for this thesis is the teleoperated robot control.

In the ISO standard, a *teleoperated robot* is defined as a robot that extends the human's sensory-motor functions to remote locations and the response of the machine to the actions of the operator is programmable. The simple way to explain it is by the etymology of the word *teleoperation*; the word comes from Greek origins where the prefix *tele* means *at a distance* and teleoperation indicates, “operating at a distance” [5]. In practice it is a technique that allows the task execution at distance and still by hand.

The main reasons for the use of teleoperation are because the direct action is not feasible or puts the life or health of the operator at risk. The operation must be performed by a human being due the unpredictability of the tasks, and for the same reason is difficult to create a model and pre program the tasks [6].

A teleoperation system is based in the master-slave architecture; with the master system situated in a control room (local environment), and the slave system situated in the remote environment, as shown in Figure 1.1. Both systems require communication and data processing due the distance between them. In the system, the human operator acts upon the master system (local task objects) and is guided with the information feedback (display) in order to execute the task through the remote manipulator [7].

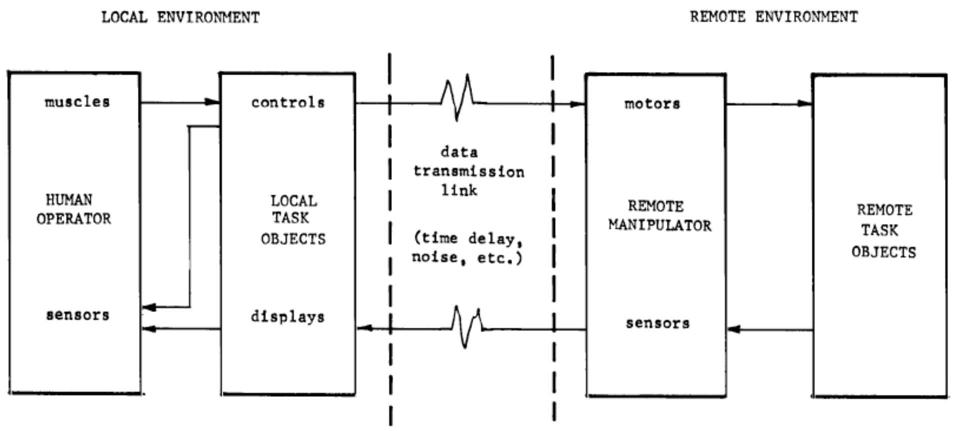



**Figure 1.2** Diagram of Remote Manipulation (from Sheridan and Ferrell, 1967)[7]

The direct and continuous interaction of the human operator with the control of the master-slave system is commonly refered as *teleoperation*. If the teleoperation system utilizes an intermediate computer as intermediary for the human operator to supervise the task, is refers as *telerobotics* [8]. Both terms can be used indistinctly, but the term *teleoperation* encompass the term *telerobotics*.

The teleoperation system can be classified in variety of ways. One of the main classifications is according to the distance to work: *short working distance* and *greater working distance*. For the systems with *short working distance,* the mechanical master-slave manipulator is a common example.

The teleoperation systems with *greater working distances* are provided by bilateral servo-controlled manipulators or motorized unilateral manipulators in open loop control without reference feedback [9].

The motorized unilateral manipulator with open loop control is limited in the adjustment of the force applied; this manipulator can't react to the environment and are commonly configured with an on-off control without force feedback. While in a bilateral servo-controlled manipulator, the main characteristic is that the system provides a force feedback to the operator. [9]

The two-way information flow between the master arm and the slave arm refers to the term *bilateral*. In a bilateral servo-controlled teleoperation system, the slave arm follows the master arm movements and the force feedback loop provides to the operator with the sense of the forces applied in the slave arm [10]. That capability of the device to provide force feedback or tactile feedback to the user is known as the haptic feedback [11].

The etymology of the word *haptics*, believed to be derived from the Greek word *haptesthai*, mean related to the sense of touch [12]. A haptic interface in the field of robotics is a device that allows human operator experiences the sense of touch in remote or simulated environments.

The common commercial haptic devices in the market are expensive, limited in the force capacity and robustness of operation. The main objective of the thesis is to study the possibility to modify an industrial robot and configure it as a haptic device for a teleoperation system.

In order to understand the decomposition of terms and the context of the definitions, the following figure is presented:

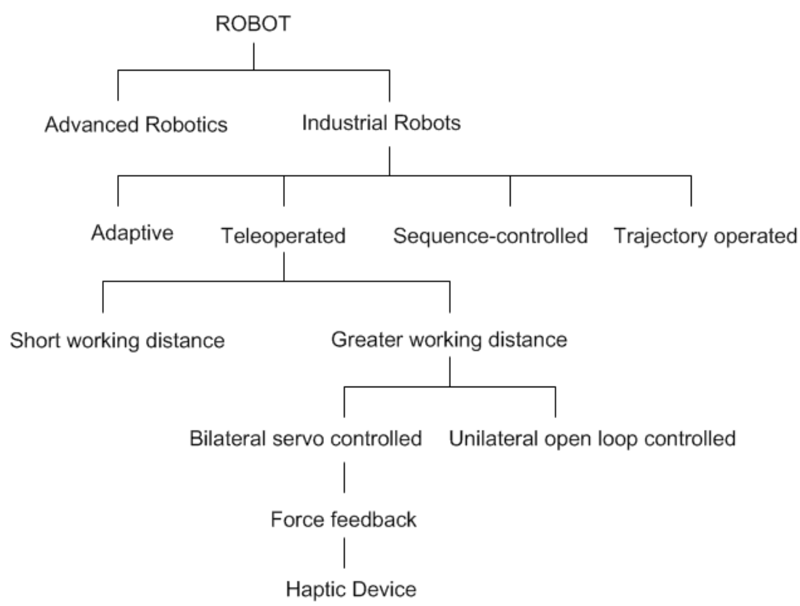



**Figure 1.3** Classification of terms and definitions

The utilization of industrial robots as the core of the teleoperation systems have been used for several decades in the nuclear industry, in applications where the main purpose is to manipulate radioactive materials in a extremely hazardous and remote environments [13]. This interface enables manual interaction with virtual environments or teleoperated slave manipulators with the advantage of force feedback. Allowing the user to manipulate and feel objects from the remote hazardous environment.

The use of off-the-shelf commercially available general-purpose robotics equipment is an alternative that reduce costs and time to develop haptic devices with the aim to simulate a large range of interaction forces [14]. The implementation of an industrial robot as input device in the teleoperation system facilitates the transformation in to a haptic device, because it is a programmable device and provides a robust performance.

That approach is the base of several research experiments [13][14][15][16]. All of those under the same commercial general-purpose industrial robot platform and with the incorporation of a 6-axis force-torque sensor mounted on the wrist of master and slave robots, due the facility to read the signal of the force and torque applied to the system [17].

One of the basic requirements in teleoperation system is the immersion[18]; where the immersion means the feeling of presence, in other words, is the sensation of being there. The force-feedback is use it to enhance the immersion of the operator in the interaction with the slave robot and facilitates perform hazardous tasks; the matching degree between the impedance transmitted to the operator from the environment impedance is called “transparency”.

The transparency is a major issue in teleoperation system [19], and in order to reach an acceptable level of transparency; the teleoperation system must be controlled in real time, where a very small latency is accepted. The real time performance is the key of the master-slave teleoperation systems with haptic interfaces.

The diversity of the skills needed to develop a teleoperation system is extensive and highly demanding in the level of expertise required. Integrating numerous fields of science and interdisciplinary applications like Electronics, Computer Science, Robotics, Signal Processing, Ergonomics, and so on. Historically, the research in teleoperation consumes a considerable amount of human capital and economic resources, through a several years of research and development, where the outcome usually is only to fulfill and specific project demand [14].

Fortunately, the robot industry is approaching a mature state where they can provide a reduction of cost in its products, with a less complexity in the programming and more extensive process documentation, and a wide product offering as well [20]. Also, the advance in computer technologies is reflected in a decrease of costs in hardware and software used for the implementation, no longer needed for big budgets in teleoperation research.

The differentiation between the actual "state of the arts" haptic devices in the market and the utilization of an industrial robot is the maximum exert able force and torque capacity, with around 22 N of torque in 6 DOF for the most advance commercial devices [21]. The range of force applied in the operation is limited, reducing the resolution of the force input, while in an industrial robots, the force input is limited for the payload capacity of the electric manipulator with a maximum of 200 N for the largest of the equipments [22].

The International Thermonuclear Experimental Reactor (ITER) is an international project which main objective is in developing fusion as a clean and sustainable energy source [24]. Inside the reactor, the fusion reaction produces high-energy neutrons, which are absorbed by the components inside the reactor vessel; therefore, a major issue for the successful operation of ITER is the maintenance of in-vessel components by remote handling methods. The Divertor Test Platform 2 (DTP2) is a project for the verification of maintenance operations of the ITER.

The Tampere University of Technology, Department of Intelligent Hydraulics and Automation (TUT/IHA) is one of the most prestige research centers, recognized as a Centre of Excellence in Research by the Academy of Finland, with an internationally high level research in hydraulics and automation [23]. Among the research activities of IHA, one of the most important is the DTP2.

The key technologies for the operation of the DTP2 are Water Hydraulics, Virtual Reality Technologies and Remote Operation. For the maintenance task of the ITER reactor a reliable remote operation is required. The resolution in the force input is one of the requirements for the teleoperation projects in the TUT/IHA.

The incorporation of accurate force sensing in the component assembly is a compulsory feature, with the objective to avoid collisions between the remote operated robot and the reactor components. The reliability of these techniques will impact in the length of the machine's shutdown phases.

The Water Hydraulic Manipulator (WHMAN) is a teleoperated manipulator arm with 6 DOF and force feedback, developed for the ITER project. With 1000 N payload when is fully extended, the WHMAN will be able to accomplish all the required tasks inside of the vessel of the ITER [25].

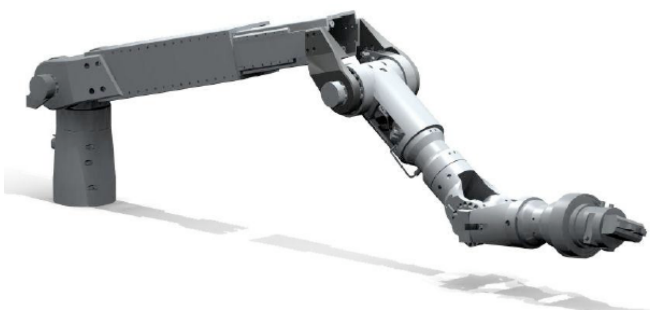


**Figure 1.4** Water Hydraulic Manipulator (WHMAN)

The dexterous manipulator configuration includes one prismatic joint and six rotary joints (hydraulic servo valves and vane type actuator). The force feedback signal in the WHMAN is acquired from the six-axis force moment sensor (JR3), mounted on the three-degree of freedom spherical wrist, and combined with the signal of pressure sensors. To measure the angular position, the feedback is acquired from dual-speed resolver in each rotary joint and LVDT-type sensor in the prismatic joint. The transducer is based in analogue technology because the high level of radiation prevents the usage of digital electronics inside the reactor.

As the master device for training tasks and operation of the WHMAN, the actual system incorporates a PHAMTOM Premium Model from the company SensAble Technologies. The device allows the user to interact with teleopearation system that require force feedback in six degrees of freedom (6 DOF), with this, the simulation of torque force feedback makes it possible to feel the collision and reaction forces and torques from the slave robot in a teleoperation environment [26]. The haptic device is connected to the PC via the parallel port (EPP) interface, supports the OS (Operation System) RedHat Fedora and the continuous exert able force and torque at nominal position is 3N.

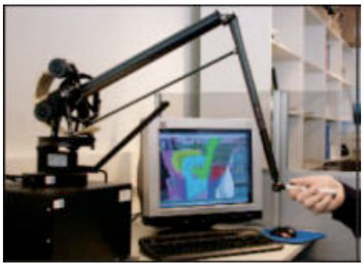


**Figure 1.5** Teleoperation through Phantom SensAble

Through the years of operation of the ITER will be necessary to manipulate and exchange components with very high loads. Water hydraulic actuators are ideal for these tasks due to the high power-to-size ratio and cleanliness of the medium in an event of leakage. Another task to perform is to cut and weld pipes in the vacuum vessel [27].

## 1.2. Motivation

The main motivation to study the field of teleoperation is the possibility to collaborate in the developing of the Water Hydraulic Manipulator. The WHMAN, from the engineering perspective, is one of the most challenging research projects and provides an attractive profile from the professional point of view. In the personal fulfillment, represents an opportunity to contribute in the research for the ITER project, a real effort to find alternatives of source of energy to reduce the global warming.

My previous studies and professional experience as mechatronic engineer are linked with the topic of this thesis and creates a continuation on the same field of studies. The motivation is to form a base of knowledge and tools for a solid background as engineer and prepare for doctoral studies.

The complexity of the teleoperation system is another motivation, due the personal commitment needed to accomplish a large project in terms of time and skills. In deed, requires the implementation of the experiment and provides a satisfaction as engineer when you see the system performance.

## 1.3. Contribution

The contribution of this thesis is to set the bases for the teleoperation system implementation, with the aim to provide another alternative to the offering of commercial haptic devices. The system based in an industrial robot modification, presents two areas of opportunity for contribute in the implementation: the robot controller and the data acquisition software.

In the industrial robot controller, the compilation of documentation and manuals, the software selection and the study of the industrial robot to configure it as the haptic device to be employed as a master arm in the teleoperation system is the contribution for the project.

For the data acquisition area, the software selection, programming and development of an interface and its communication with the robot controller is the main contribution, because the software is not directly supported and there is not documentation for the configuration.

Another contribution is the analysis of previous teleoperation system and the study of the state of the arts systems in this field, in order to find alternatives for reach the require performance.

Finally, the combination of industrial robot and the virtual technologies is an advantage in the implementation of the project. Integrating and testing the whole project in a computer simulation reduces the implementation time and the possible failures in the system. Taking advantage of the simulation software from the robot company manufacturing and from a flexible data acquisition software of a third part software developer is possible to accomplish the settled goals in a reasonable period of time.

# 2. BACKGROUND

The objective of this chapter is to present the description of the basic components in the teleoperation system in order to understand the context of the research. The content is linked with the requirements to develop a haptic interface for the system presented in the introduction and highlights the functionality of each component. The level of detail in the documentation is not extensively describing it due to the amount of components involved in the teleoperation system; therefore the reader should have a background in the topic.

Focusing in the components of the master arm, the system could be divided into two main areas: the robot controller and the data acquisition platform. Both areas are complex and compounds of multiple software and hardware, in order to enhance the performance of the interaction of the system throughout the operation and communication, every minimum detail in the components performance should be taken in to account, with the aim to avoid bottlenecks in the system.

## 2.1. System Architecture Description

The architecture of the system is according to the requirements for the configuration of a master arm device, its programming, modification and the communication for the data acquisition. The system layout is presented in the figure 2.1, where the three basic hardware components are: the industrial robot manipulator, the robot controller and the computer. The hardware and software required for the operation will be described in this chapter.

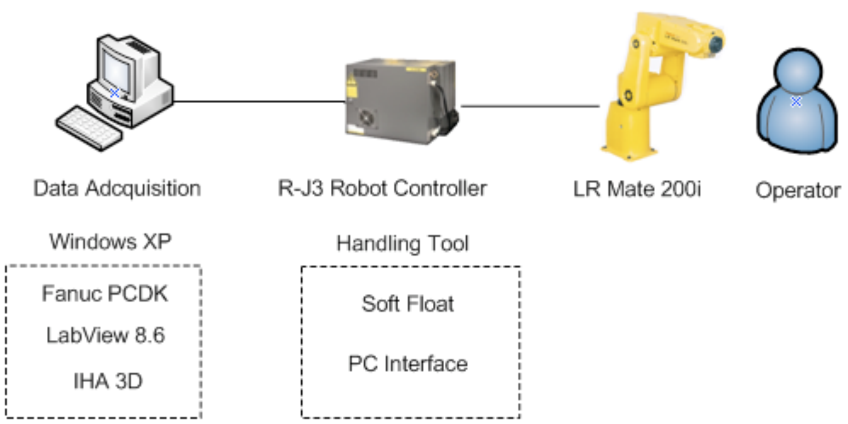



**Figure 2.1** Teleoperation system components

## 2.2. The Industrial Robot Selection

The estimation of industrial robots in use is around 1 million robots worldwide, where the leader country in stock and sales of multi-purpose industrial robot is Japan, and United States in second place with more than 188,000 units. Base it on those statics; the main robot manufacturers according with robots installed worldwide are FANUC Robotics, Motoman Inc and ABB Inc [28].

FANUC Robotics is the leader in the industry with over 200,000 installed robots and its own global customer service network. The products of the company include 200 robot variations and a range of 0.5 kg to 1200 kg payload [29]. This characteristic is what is made a suitable robot election for the selection of the robot manufacturer.

Among the product of Fanuc Robotics we can find articulated robot structures with a payload in the range of the needed for the resolution of the force input, and the LR Mate 200 Series is the most suitable option for the teleoperation system. The LR Mate 200 Series also called anthropomorphic manipulator has a human-like form, that manipulates mechanical objects with what resemble human body motions and provide the operator with a remote body image or physical alter ego [8]. The reason to select this robot series is the suitable size of the model to be used as master device and its kinematics similarity with the WHMAN. The characteristic of the robot and the controller are described in the following paragraphs.

### 2.2.1. LR Mate 200i Robot Arm

The LR Mate 200i is a 6-DOF industrial robot, manufactured by FANUC Robotics and designed for a variety of manufacturing and system processes. The manipulator provides a 30 N of payload capacity in a compact modular construction and a reach of 700mm with enough flexibility and higher reliability through the harmonic drives in its six-axis.

The robot feedback position use absolute serial encoders in order to eliminate the need for calibration at power-up and the motors are brushless AC servomotor with the objective to minimize the motor maintenance. The repeatability is +/- .04mm and the axes speeds is up to 480° per second. The end-effector connector built into the wrist facilitates the connection of inputs and outputs in the tip of the manipulator[30].

**Figure 2.2** The LR Mate 200i (Courtesy of Fanuc Robotics)[30]

For security, the robot have a fail-safe brakes on axes 2 and 3, and I/O module for implement safety hardware as a safe guards. The robot is available with R-J2 Controller for basic material handling or R-J3 Controller for complex process applications. The controller for this implementation is the R-J3 Controller.

### 2.2.2. R-J3 Controller

The R-J3 is a Fanuc Robotics third generation robot controller, designed with process capability and open architecture features that improves application and motion performance [31].

The controller processor architecture uses a 32-bit main CPU with dual processor that permits fast calculations, reduces program execution times and increases path accuracy. The system support auxiliary axes with its own control program and simple kinematic models.

Another process features are collision detection that minimizes potential damage to the robot and Turbo Move that provides minimal cycle time by computing robot dynamic in real time. Among the communications the networking capabilities include built-in Ethernet interface and RS-232 ports.

**Figure 2.3** R-J3 Controller (Courtesy of Fanuc Robotics)[31]

In the Figure 2.3 the visualization of the controller box, the user interface and the teach pendant for programming and configuration of the controller. One of the software for the controller configuration is Handling Tool software with specially purposes features for handling operation.  The software is designed for manipulating work pieces through the operation of the LR Mate 200i robot manipulator.

### 2.2.2.1. Handling Tool Software

Handling Tool is the software installed on the R-J3 control unit for material handling operation; this tool contains instructions for controlling the robot, hands, remote control units, and other peripherical devices.

The software enables the following features: setting up the system, creating a program, testing the operation and status display for monitoring [32]. When optional functions are incorporated, the system can be expanded and the management functions can be enhanced.

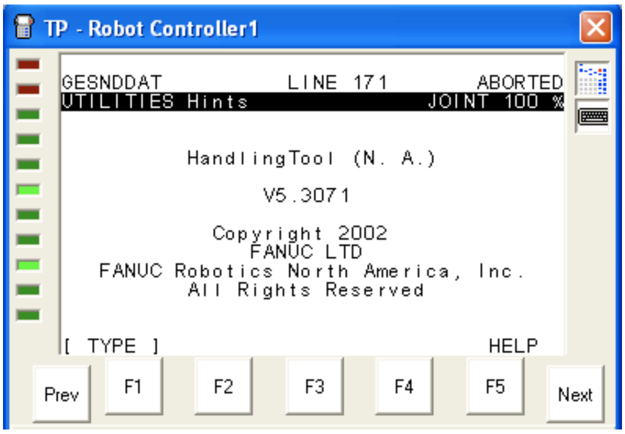



**Figure 2.4** Handing Tool main window (from Fanuc Roboguide)

The Handling Tool supports a variety of programs that enhance the motion performance of the robot; one option for increase the flexibility of the motion in the controller is the option named Soft Float.

Soft Float is used in processes to compensate for unexpected variances, due to the occurrences of these variances; Soft Float allows the robot path to be changed according to the external force applied to achieve the desired result [33]. The detailed functionality of this software option is presented in the next section.

Another feature in the Handling Tool software is the Host Communication screen, where the protocols of communication supported are displayed. For a RJ3 controller, the basic communication protocols are: TCP/IP, TELNET, Socket messaging, Point-to-Point and FTP File Transfer Protocol.

The TCP/IP (Transmission Control Protocol (TCP) and Internet Protocol (IP)) is the most common protocol linked with Ethernet and the first option for data transmission in this implementation [34]. In order to configure the communication, the installation of the PCIF option is required. The PC Interface (PCIF) is an option in the RJ3 controller that enables the communication within the Personal Computer (PC) and the robot controller unit. More detailed information is presented in the section 2.2.2.3.

### 2.2.2.2. Soft Float

Soft Float is an option of the Handling Tool supported in the RJ3 Controller and used in processes to compensate for unexpected variances. Usually needed to mount workpieces on a machine tool, if exist variances in the pieces that cause interference, the robot path can be changed according to an external force to achieve the desire position.

The functionality is that the robot detects the external force and can follow to the force direction. For example, if the robot is unloading a work piece from die-cast machine, the robot handles the work piece ejected in a flexibly way.

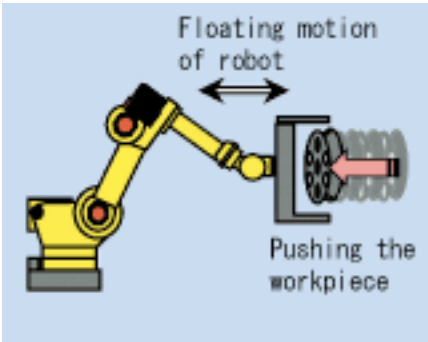



**Figure 2.5** Cartesian Soft Float application (Courtesy of Fanuc Robotics) [29]

Exist two types of Soft Float modes: Joint and Cartesian [35]. In Joint mode, flexibility is specified for individual axes or a combination of axes. In this mode, the robot will act like a spring in the specified direction. For Cartesian mode, the softness is specified for Cartesian directions.

The detailed setup and configuration of the screens for both modes is presented in the next chapter.

### 2.2.2.3. PC Interface

In the system R-J3 Robot Controller, the built-in Ethernet interface is an option of communication, with the standard 10BaseT. However, the option is not part of the base configuration of the controller and should be ordered by separate. The option that enables this communication is the PC Interface (PCIF).

PC Interface software allows Fanuc robots to exchange data with a PC-based FANUC Robotics application software package or a custom application created with PC Developers Tool Kit. The features of the software are described in the next paragraphs.

## 2.3. Data Acquisition System

In this thesis the data acquisition system is referred as the manipulation of the position and torques values, read from the robot controller. This data is collected in a PC by the tools provided from Fanuc Robotics System (PC Developer Kit software) and presented in an interface programmed in LabView. After manipulation, the data are send to the IHA3D virtual environment for visualization of the interaction with the remote environment and the feedback to the robot controller.

The figure 2.5 shows the flow of information between the different software components of the data acquisition system. A detailed description is presented in this section.

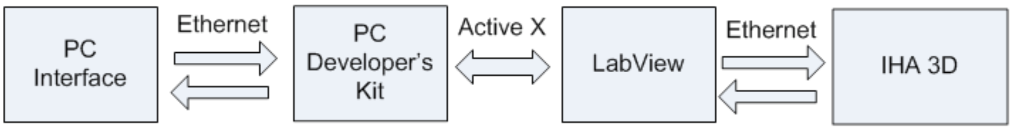



**Figure 2.6** Data acquisition interaction

### 2.3.1. PC Developer's Kit

PC Developer's Kit is a software tool that provides a bridge for communicates the RJ3 controller with the Windows environments through the use of a set of libraries with predefined commands. PCDK enables high performance communication of information and instructions between a PC and FANUC Systems. The kit is both a development and run-time environment that gets the MS Windows application running quickly.

**Figure 2.7** PC Developer's Kit interface (Courtesy of Fanuc Robotics)[29]

The kit is compound of two main programs: Robot Server and Robot Manager. The first one is used to access variables directly with a single R-J3 Robot Controller, while the second one is used for controlling a monitoring several connections between Robot Server and the R-J3 controller. PCDK contains an example program for both, a FRRobotDemo for Robot Server and FRRobManDemo for Robot Manager.

FRRobotDemo is a useful guide to understand the functionality and possibilities of PCDK and the source code is provided as example. The source code is a program in Visual Basic 6 and the set of libraries of PCDK is based in ActiveX framework and specifically talking is a collection of .DLL files. The program used as reference for this implementation is Robot Server.

#### 2.3.1.1. Robot Server

Robot Server is an Active-X executable program that provides to the PC application access to an R-J3 robot controller. It communicates with the robot using TCP/IP (typically over Ethernet) and to the PC application through an object (COM) interface.

And important issue is the each new version of the Robot Server supports all of the previously supported R-J3 controller versions and features. However, the reverse is not true. The older versions of the Robot Server are not guaranteed to work with newer versions of the R-J3 controller.

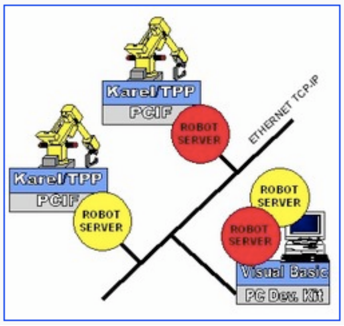



**Figure 2.8** PC Developer's Kit interface (Courtesy of Fanuc Robotics)[29]

FANUC Robotics Robot Server and PC applications interaction is based in the communication with the R-J3 controller through the performing of the following operations: Read/write variables, monitor/set I/O, coordinate with user written robot programs, Monitor alarms, Control programs, Modify positions.

These protocols integrate seamlessly with the Visual Basic IDE and are launched automatically when your application creates an instance of the robot controller.

## 2.3.1.2. ActiveX

In order to create a program for communicate with the R-J3 controller the understanding of Active X controls is necessary. Active X is a framework for defining reusable software controls that perform a particular function or set of functions in Microsoft Windows and is independent of the programming language used to implement them.

Due the fact that PCDK is based in Active X framework and utilizes library of controls Dynamic-link Library (.DLL) a Visual Basic programming expertise is required to develop application packages. Visual Basic is an event driven programming language and integrated development environment (IDE) from Microsoft for its COM programming model. Visual Basic is also considered a relatively easy to learn and use programming language, because of its graphical development features.

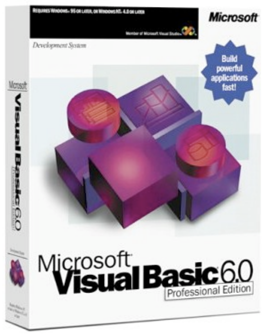



**Figure 2.9** Microsoft Visual Basic 6.0 (Courtesy of Microsoft)

The PCDK help documentation is completely based to work with Visual Basic, however with the aim to simplify the data acquisition for the implementation, the utilization of LabVIEW as the software for programming the control is selected. LabVIEW is more flexible and visual to program and supports the Active X control interaction.

### 2.3.2. LabView

LabVIEW is a graphical system design software platform for control, test, and systems development using intuitive graphical icons and wires that resemble a flowchart. A complete program is developed in this platform for acquire position and torque values from the RJ-3 Robot Controller.

The LabVIEW interface displays the actual position of the robot joints based on the pulse count from the absolute encoder and the disturbance torques values acquired from the robot controller. The values are manipulated and presented in a suitable format for the understandings of the operator and at the same time are transmitted to the IHA3D software for the visualization of the manipulator behavior over the virtual environment.

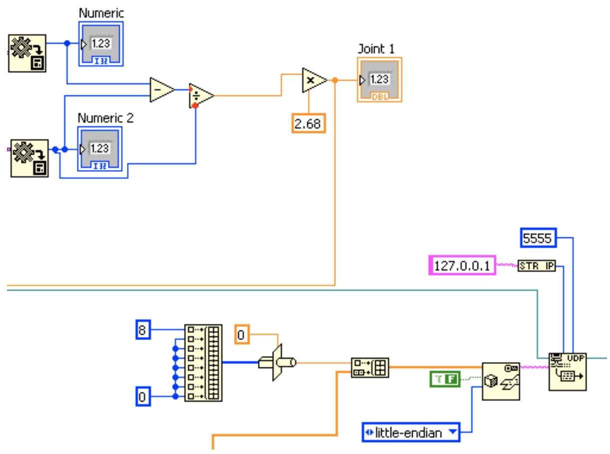



**Figure 2.10** LabVIEW 8.6 programming code example.

The communication protocol used between LabVIEW and the R-J3 Robot Controller is TCP/IP, while the communication protocol between LabVIEW and IHA3D is UDP/IP. A detailed explanation of the program is presented in the chapter 4.

## 2.4. Virtual Environment

The virtual environment is defined as the simulation of the robot system in a computer. With the objective to enhance the immersion of the operator, the robot and the interaction with the environment is rendering in a graphical interface. A computer monitor shows the movements of the robot in simulation software specific designed for this purpose. The LR Mate 200i could be simulated in Roboguide while the WHMAN is simulated in IHA3D software; the characteristics of both are explained in the next points.

## 2.4.1. Roboguide

Fanuc Robotics ROBOGUIDE is an off-line robot simulation software, with a Virtual Robot Controller the user can design and test a robot system in a 3D space prior to delivery of the physical robot.

The software allows the import of CAD models of parts, tools machines and work cells. Roboguide includes an Integrated Virtual Teach Pendant that looks and operates like a real teach pendant and facilitates the simulation of the operation and performance of a robot system and the evaluation of the cycle times and reach.

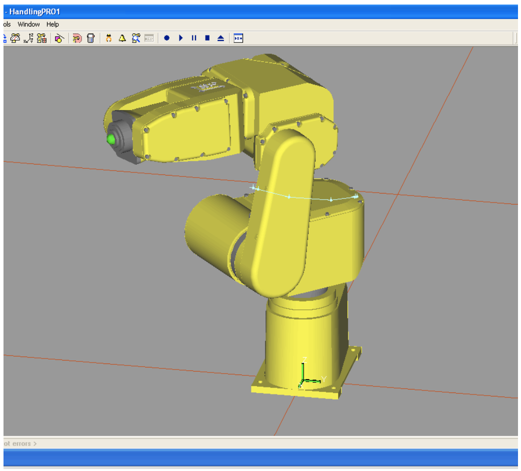


**Figure 2.11** ROBOGUIDE virtual environment (from Fanuc Roboguide)

## 2.4.2. IHA3D

IHA3D is a visualization environment that creates the virtual scenario and the robot movements on line. The software is used to display the simulation of the WHMAN with the objective to visualize the orientation of the robot and its movements. For the coordination of the movements, the system utilizes the Tool Frame and the Reference Frame to determine the kinematics of the robot.

To accomplish the teleoperation system the IHA3D allows building complex 3D environments and support sending and receiving information via network connections. The protocol for the implementation of the incoming and outgoing network connections is User Datagram Protocol (UDP), this due the simplicity of the transmission model without implicit hand shaking dialogues for guaranteeing reliability, ordering, or data integrity. But in the other hand increase the speed of the communication transmission and the performance of the system.

Another important feature of the software is collision detection; simulating the possible scenarios inside the reactor, the software provide feedback when any of the robot links crash with the virtual predefined parts in the 3D environment providing another tool to avoid damage to the reactor and the robot itself.

The following figure shows the visualization of the WHMAN in the IHA3D environment and the frame visualization.

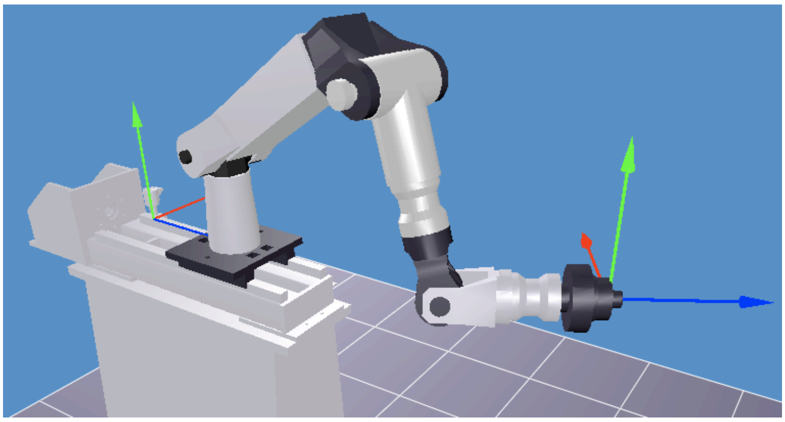


**Figure 2.12** IHA3D virtual environment

An important property of the IHA3D software is it was designed in the Tampere University of Technology specifically for this purpose and provides a complete flexibility in terms of modifications of the core programming code and its on evolution is related with the ITER project. A more complete explanation of the implementation is presented in the next chapter.

# 3. RESEARCH METHODS

The methodology used for develop this thesis is based in an empirical research; understood as the use of direct observation or experiment. In which the experiment for the research involves the implementation of the teleoperation system based in a master arm and its interaction with the virtual environment.

For this implementation, overcome the limitations of work over an off-the-shelf equipment platform is a key issue, and the first challenge is to setup the industrial robot to move it freely and following the movement of the operator hand. That configuration we can call it “force control”, and in order to setup the force control mode, we have to understand how the robot works internally, utilize the configuration screens and the already programmed software options.

The Handling Tool software provides a set of screens for setting up the system for material handling applications. However, if we want to overcome the limitations and extend the functionality of the software architecture provided from FANUC the modification should be do it directly to the variable of the controller. This means that the information published in the manuals to configure the robot is not enough base of knowledge for this research. Therefore is considered that the researcher understands and knows how to operate and program a R-J3 controller and is familiarized with the hardware structure of the LR Mate 200i.

Usually an industrial robot moves according to a jog feed specified on the teach pendant, for this implementation the objective is that the robot move according to the movement of the hand of the operator. The normal functionality of an industrial robot differs from the functionality needed when is configured as a master arm for a teleoperation system, there is no move instruction introduced in a program to execute and move to the target position specified in a jog feed. This is a paradigm when the engineer is unfamiliar with the teleoperation field. The figure 3.1 shows the WHMAN manipulation by the operator through the use of the LR Mate 200i.

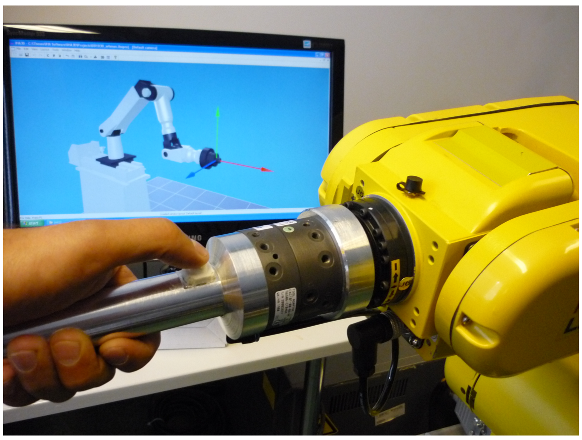


**Figure 3.1** LR Mate200i manipulation

In order to setup the robot controller to be able to move the end-effector freely by the operator hand, the robot controller offers the possibility of utilize the Soft Float option. The detailed functionality and configuration is shown in the next point.

## 3.1. Soft Float Configuration

The basic functionality of a robot is to move from one position to another position previously specified by a teach pendant. If we want to move the robot freely in the workspace and synchronize it with the hand movement, the Soft Float function is the solution.

Through Soft Float the robot detect the applied external force and can follow the force direction. This functionality supports two types of configurations: Joint Soft Float and Cartesian Soft Float. The first one works specifying the softness related to the direction of rotation of each arm of the robot, and the second one works specifying the softness on the Cartesian axes. The Soft Float function is enabled or disabled by using an instruction in the program. This create the necessity to execute a program with the next structure:

```
1: J  P [1]   100%   CNT100
2:     SOFTFLOAT [1]
3: J  P [2]    100%   CNT100
4:     FOLLOW UP
5: J  P [3]    100%    CNT100
6:     SOFTFLOAT  END
```

Once that Soft Float is enable is possible to move the robot with the hand, then the next step is to define the amount of force needed to move it. The servo flexibility indicates how strongly the axis resists external forces and the parameter can be specified for each axis. Through the use of Soft Float screen interface, the servo flexibility could be specified between 0% and 100% configured over the teach pendant, where a servo flexibility of 100% corresponds to being the most flexible. If we want to increase the level of servo flexibility, the configuration should be done writing directly to the variable.

The setup of the servo flexibility must be calculated according with the force feedback signal taken from the slave arm. Taken into account that if the force applied in the master arm by the operator is above the level of the servo flexibility (so high as to overcome a static frictional force, see figure 3.2), the axis of the robot is pressed and moved. So when a static load is applied to the robot, the robot controls force to maintain its attitude. The servo flexibility is specified using the condition tables presented in the following points.

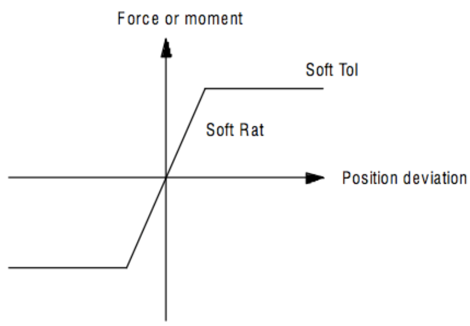



**Figure 3.2** Soft Ratio and Soft Tolerance Graphic (from FANUC Robotics)[32]

### 3.1.1. Joint Soft Float

The figure 3.3 shows the condition table in the interface of the Joint Soft Float. Through the screen interface the maximum value that is possible to introduce is 100 % corresponding to be the most flexible. However, in the table 3.1 the correspondent variable shows that the range of input for the servo flexibility is between 0 and 255. In a practical application, the maximum value for the servo flexibility in Joint Soft Float is 127 %, because after introduce a greater value, the controller starts taking the values as negative and cause failure in the mathematical calculation.

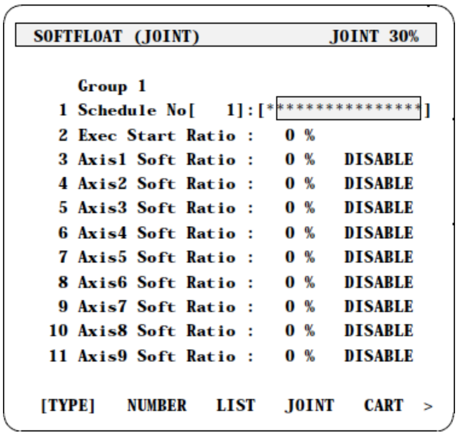



**Figure 3.3** Joint Soft Float Screen (from Handling Tool)

| | Screen Name | Variable Name | Type | Min | Max |
|---|---|---|---|---|---|
| 3 | Axis1 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[1] | Byte | 0 | 255 |
| 4 | Axis2 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[2] | Byte | 0 | 255 |
| 5 | Axis3 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[3] | Byte | 0 | 255 |
| 6 | Axis4 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[4] | Byte | 0 | 255 |
| 7 | Axis5 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[5] | Byte | 0 | 255 |
| 8 | Axis6 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[6] | Byte | 0 | 255 |
| 9 | Axis7 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[7] | Byte | 0 | 255 |
| 10 | Axis8 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[8] | Byte | 0 | 255 |
| 11 | Axis9 Soft Ratio: | $SFLT_GRP1[1].$SFLT_VAR[9] | Byte | 0 | 255 |
| 3 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[1] | Boolean | False | True |
| 4 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[2] | Boolean | False | True |
| 5 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[3] | Boolean | False | True |
| 6 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[4] | Boolean | False | True |
| 7 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[5] | Boolean | False | True |
| 8 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[6] | Boolean | False | True |
| 9 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[7] | Boolean | False | True |
| 10 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[8] | Boolean | False | True |
| 11 | Disable/Enable | $SFLT_GRP1[1].$SFLT_ENB[9] | Boolean | False | True |

**Table 3.1** Joint Soft Float correspondent variable and values

The behavior of the robot in Joint Soft Float is not desired for the force control mode. In that mode, the robot acts like a spring in the direction of the force applied. If the servo flexibility overcomes certain value, the spring behavior decrease but remains moving in small steps. That behavior can be avoided setting up the robot controller in Cartesian Soft Float.

### 3.1.2. Cartesian Soft Float

The figure 3.4 shows the condition table of the interface for the Cartesian Soft Float. The behavior is according with the graphics presented in the figure 3.2 where the direction and rotation for the servo flexibility are configured with the values of the Soft Ratio and Soft Tolerance. As well as the previous interface, the maximum input value for both parameters are 100% in the software interface or if we write the value directly to the variables the range is between 0 and 255.

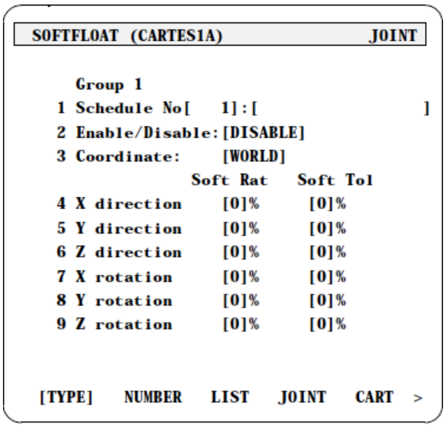



**Figure 3.4** Cartesian Soft Float Screen (from Handling Tool)

| | Screen Name | Variable Name | Type | Min | Max |
|---|---|---|---|---|---|
| 4 | X direction Soft Rat | $SFLT_GRP1[1].$CSF_KX | Integer | 0 | 255 |
| 5 | Y direction Soft Rat | $SFLT_GRP1[1].$CSF_KY | Integer | 0 | 255 |
| 6 | Z direction Soft Rat | $SFLT_GRP1[1].$CSF_KZ | Integer | 0 | 255 |
| 7 | X rotation Soft Rat | $SFLT_GRP1[1].$CSF_KU | Integer | 0 | 255 |
| 8 | Y rotation Soft Rat | $SFLT_GRP1[1].$CSF_KV | Integer | 0 | 255 |
| 9 | Z rotation Soft Rat | $SFLT_GRP1[1].$CSF_KW | Integer | 0 | 255 |
| 4 | X direction Soft Tol | $SFLT_GRP1[1].$CSF_FXLIM | Integer | 0 | 255 |
| 5 | Y direction Soft Tol | $SFLT_GRP1[1].$CSF_FYLIM | Integer | 0 | 255 |
| 6 | Z direction Soft Tol | $SFLT_GRP1[1].$CSF_FZLIM | Integer | 0 | 255 |
| 7 | X rotation Soft Tol | $SFLT_GRP1[1].$CSF_NULIM | Integer | 0 | 255 |
| 8 | Y rotation Soft Tol | $SFLT_GRP1[1].$CSF_NVLIM | Integer | 0 | 255 |
| 9 | Z rotation Soft Tol | $SFLT_GRP1[1].$CSF_NWLIM | Integer | 0 | 255 |
| | | | | | |

**Table 3.2** Cartesian Soft Float correspondent variable and values

In practical application, the maximum value recommended for the parameter of Soft Ratio is 103%, after this value, the behavior of the robot tends to loose its stability and increase its force of inertia, causing a possible damage to the operator and the robot itself. Due possible safety problems, the strategy is to configure the robot parameters needed to safeguard the integrity of the operator.

### 3.1.3. Soft Float Security and Conditions

The operation of Soft Float must fulfill certain conditions to function, as the teleoperation system requires. If the robot controller displays the message error “SRVO-023 SERVO Stop error excess (G:1 A:1)”. The cause is an excessive servo positional error, due the load applied.

The possible solution is an increment of the joint limits according to the next Soft Float Error limits for the variable “$PARAM_GROUP[1].$SFLT_ERLIM”, where the variable type is integer and the minimum value is 0 and the maximum value is 100,000,000 correspondent to the angle of the encoder.

| Old Values | New Values |
|---|---|
| [1] 2621440 | 83886080 |
| [2] 2621440 | 83886080 |
| [3] 2621440 | 83886080 |
| [4] 5242880 | 83886080 |
| [5] 5242880 | 83886080 |
| [6] 5242880 | 83886080 |
| [7] 5242880 | |
| [8] 5242880 | |
| [9] 5242880 | |

The new values increase the range of operation for Soft Float, approximately three times. However if we want to have a level of security in the test at the moment of increase the servo flexibility over 103% the recommendation are the next values:

$PARAM_GROUP[1]. $SFLT_ERLIM [1]

| | |
|---|---|
| 1 [1] | 10485760 |
| 2 [2] | 5242880 |
| 3 [3] | 5242880 |
| 4 [4] | 10485760 |
| 5 [5] | 5242880 |
| 6 [6] | 5242880 |
| 7 [7] | 524288 |
| 8 [8] | 524288 |
| 9 [9] | 524288 |

Once that Soft Float is configured, the robot can be moved according with the movements of the hand of the operator, and the next step is to configure the communication with the computer.

## 3.2 Host Communication Configuration

The setup of the communication between the robot and the computer is through the parameters of the Host Comm screen. The Host Comm screen is shown in the figure 3.5 and displays the protocols supported in the R-J3 Controller for communication.


**Figure 3.5** Host Comm Screen (from Roboguide)

After select the protocol TCP/IP, the software shows the screen for the configuration of the protocol, see figure 3.6. The interface facilitates the setup of the communication where the parameter "Node name:" is the name of the robot controller and should be identical to the parameter "Host Name (LOCAL)". While the Internet Address specifies the IP address of the robot. If the system doesn't use a router, the parameter "Router name:" can be left in blank.

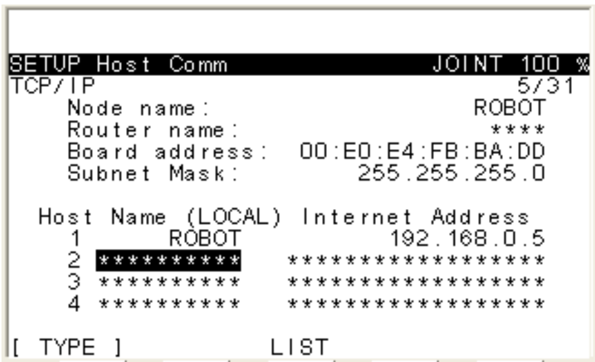



**Figure 3.6** TCP/IP configuration screen (from Roboguide)

Once that the IP address is assigned and the server is started, the robot controller is ready to communicate with the computer. Is important to remember that the PC Interface (PCIF) option should be installed in the controller in order to establish the communication. The next step is to configure the communication from the computer side.

## 3.3. Data Acquisition System Programming

The minimum system requirements for communicate the computer with the R-J3 Robot Controller is to install the PC Developer's Kit with the FANUC Robotics Robot Server. The Robot Server is an Active-X executable program that provides access communication to LabVIEW with the R-J3 Robot Controller through TCP/IP.

The PCDK documentation was designed to guide and develop programs based in Visual Basic IDE and the robot library is launched automatically through the .DLL protocols. The knowledge of Visual Basic programming skill is important for programming purposes, however is not indispensable due the compatibility of controls and functions of LabVIEW and it's supports access to Active-X technology.

The lack of documentation from FANUC oriented to the integration of LaVIEW and its robot controllers justifies publish the programming code in this chapter. LabView is the core programming language where the communication is established and the computer interface is created for the implementation and the program is detailed explained with the objective to create a knowledge base for future integrations.

### 3.3.1. Connection and Variables Access

The Robot Server software utilizes the object-oriented method to access the data into the R-J3 Controller. The source is obtained from the object named "Robot Object" that is the interface to the Robot Server and the same time the window into the R-J3 Controller. In the Robot Server we can found a variety of object classes labeled with the prefix FANUC Robotics Class (FRC). In this context, the class name for the "Robot Object" is FRCRobot and can be referenced in the library "FANUC Robotics Controller Interface Version 1.5".

LabVIEW provides a set of functions for connectivity with Active-X, the figure 3.7 shows the basic program for establish the connection and the table 3.3 explains the function of each control.

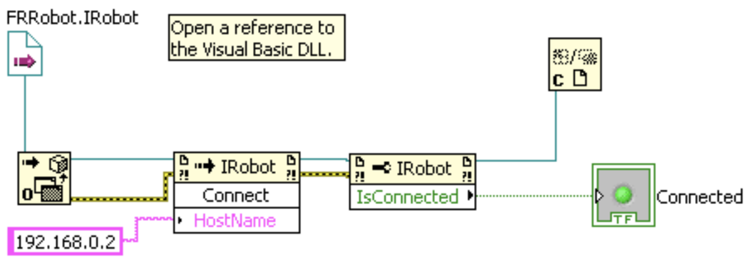



**Figure 3.7** LabVIEW program for establish a connection

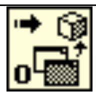
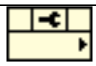
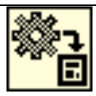
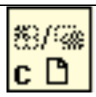
| Icon | Name | Function |
|---|---|---|
|  | Static Reference | Reference to the library “FANUC Robotics Controller Interface Version 1.5” and the Active-X class “IRobot”. |
|  | Automation Open | Returns an automation reference number, which points to a specific Active-X object. |
|  | Invoke Node | Invokes the “Connect Method” that connects to the robot controller specified by Host Name. |
|  | Property Node | The node automatically adapts to the class of the object. The property “IsConnected” returns a Boolean flag that can be used to determine if the Robot Object is connected to a robot controller. |
|  | Variant to Data | This function converts “variant data” to a LabVIEW data type, then LabVIEW processes and displays the data. |
|  | Close Reference | This function closes a reference number associated with an open Active-X object. |

**Table 3.3** LabVIEW function for Active-X connectivity

Once that the connection is establish, the next step is get access to the variables. The Robot Object utilizes a hierarchical method where each property returns an object class that is pointing in the next level of detail. The example for access a variable is shown in the figure 3.8 and the figure 3.9 shows the program example for access I/O of the robot in LabVIEW.

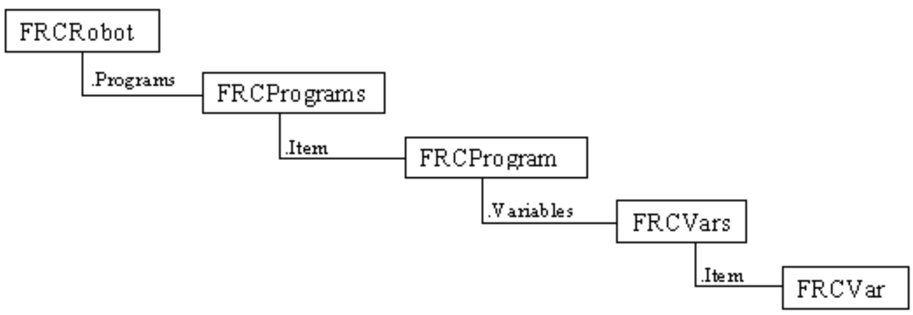



**Figure 3.8** Hierarchical View for access variables (Courtesy of FANUC Robotics)

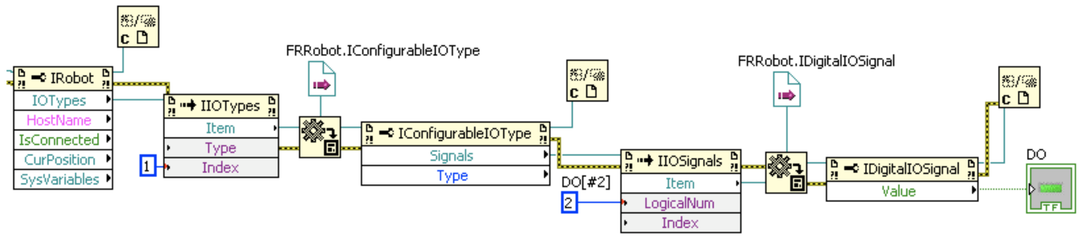



**Figure 3.9** LabVIEW program for access robot digital I/O

The most common method for obtain data from the robot controller is through the "FRCVars" and "FRCVar" class objects. The "FRCVars" collection is an "FRCVar" object from where the variable can be accessed. The "FRCVar" object has the "Value" property that allows you to read and set the variable value. For example, the figure 3.10 shows the program for access the value of the variable "$SPC_COUNT[1]" and display it in a numeric value.

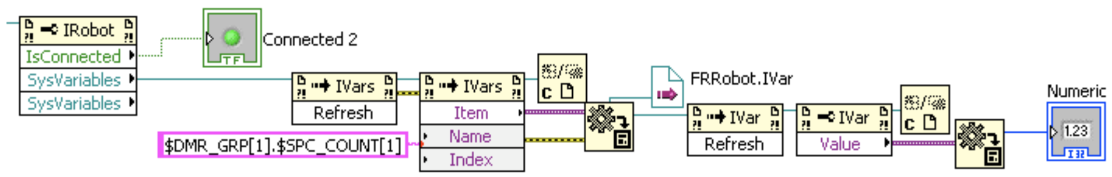



**Figure 3.10** LabVIEW program to access absolute encoder count

Now that is possible to establish a connection, monitor, read and change variables in the controller, the next step is to know what are the variables that are relevant for the implementation of the teleoperation system.

# 4. IMPLEMENTATION

The Robot Object has different format types for position coordinates. For the R-J3 Controller the supported position coordinates are Joint representation, Jog Frame or Transform and XYZWPR format. The figure 4.1 shows the hierarchy to access the types of position variables, but is important to know that in the position variables after the value has been read from the controller, the value is not updated. If we want to update the value, the program should call the Refresh method or the Update method.

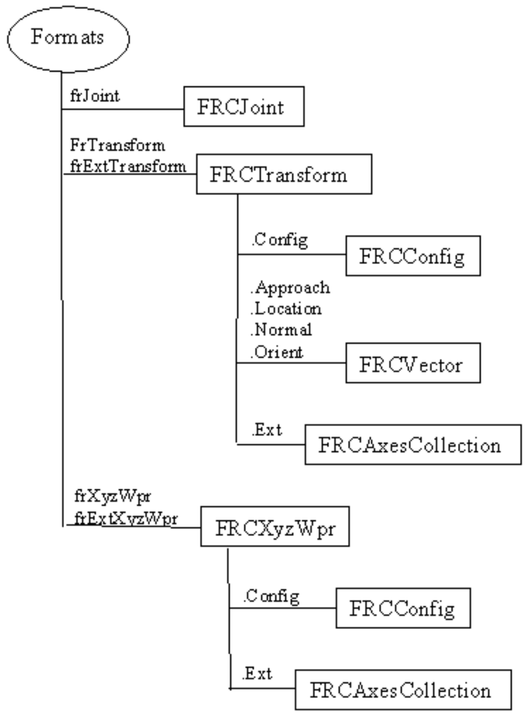



**Figure 4.1** Access hierarchy for position variables (Courtesy of FANUC Robotics)

Another way to obtain the position value for each axis is through reading the values from the absolute encoders variable ("$DMR_GRP[1].$SPC_COUNT[1]"). The system variable "$SPC_COUNT[9]" is the value from the Serial Pulse Coder Count, where the system stores the encoders count value every basic motion cycle.

The encoder count value can be subtracted to the value of the master count data of each joint stored in the variable "$DMR_GRP[1].$MASTER_COUN[1]" and then divided with the number of pulses per revolution of each encoder to get the value of the position in Joint coordinate system.

For the teleoperation system the use of Cartesian coordinate system is required due the differences in the mechanical structure between robots. To convert a Joint coordinate system into a Cartesian coordinate system (XyzWpr), the computation of the forward kinematics is needed.

## 4.1. Forward Kinematics

The aim of calculates the forward kinematic is to get the position and orientation of the end-effector as a function of the joint variables [36]. The position and orientation of the robot in the standard Cartesian coordinate system is fixed in a workspace or in a position determined by the robot. The workspace and dimension of the LR Mate 200i are shown in the next figure.

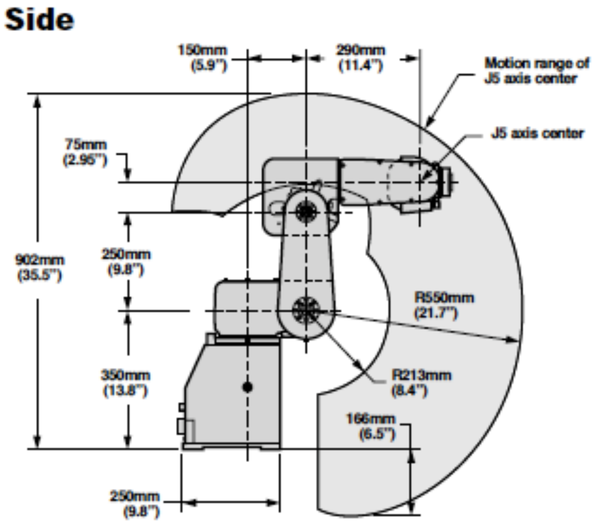



**Figure 4.2** LR Mate workspace side view (Courtesy of FANUC Robotics)[30]

To calculate the forward kinematics of the open-chain manipulator, the Denavit-Hartenberg convention was adopted and the frame assigning was according to the figure 4.3. The description of the convention is the next one:

$a_i$ = distance between $O_i$ and $O_{i'}$,

$d_i$ = coordinate of $O_{i'}$ along $z_{i-1}$

$\alpha_i$ = angle between axes Zi-1 and Zi about axis xi to be taken positive when rotation is made counter-clockwise

$\vartheta_i$ = angle between axes xi-1 and xi about axis zi-1 to be taken positive when rotation is made counter-clockwise

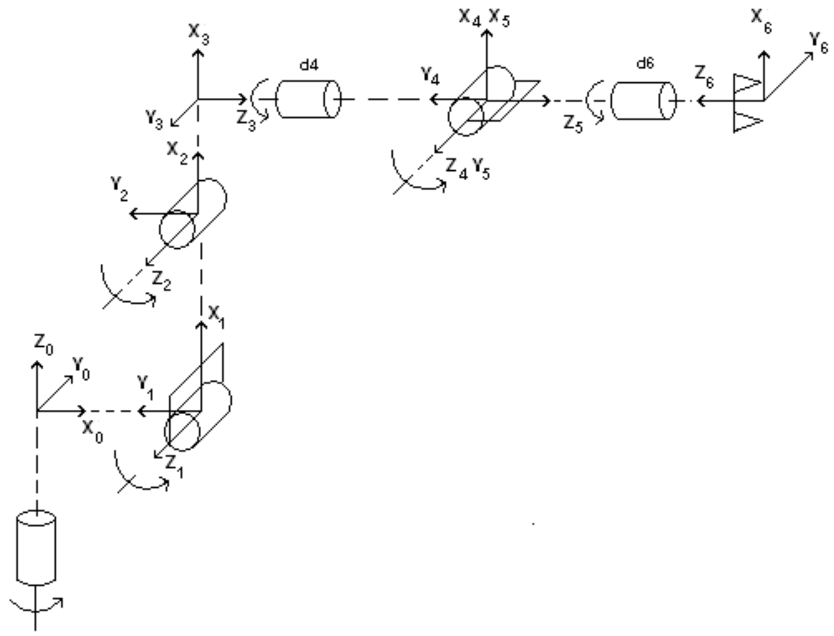



**Figure 4.3** Frame assigning of the manipulator

The frame assigning corresponds to the parameters utilized for calculate the forward kinematics in the robot controller; this minimizes the possible errors in the calculation of the Cartesian coordinates. The table 4.1 shows the Denavit-Hartenberg parameters of the LR Mate 200i.

| | $a_i$ | $\alpha_i$ | $d_i$ | $\vartheta_i$ |
|---|---|---|---|---|
| 1 | 150 | π/2 | 0 | $\vartheta_1$ |
| 2 | 250 | 0 | 0 | $\vartheta_2$ |
| 3 | 75 | -π/2 | 0 | $\vartheta_3$ |
| 4 | 0 | π/2 | -290 | $\vartheta_4$ |
| 5 | 0 | -π/2 | 0 | $\vartheta_5$ |
| 6 | 0 | π | -80 | $\vartheta_6$ |

**Table 4.1** Denavit-Hartenberg parameters

The coordinate transformation is the result of the multiplication of the single transformation:

$$\boldsymbol{A}_i^{i-1}(q_i) = \boldsymbol{A}_{i'}^{i-1}\boldsymbol{A}_i^{i'} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i}c_{\alpha_i} & s_{\vartheta_i}s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i}c_{\alpha_i} & -c_{\vartheta_i}s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^0(\vartheta_1)=\begin{bmatrix} c_1 & 0 & s_1 & a_1c_1 \\ s_1 & 0 & -c_1 & a_1s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_2^1(\vartheta_2)=\begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2(\vartheta_3)=\begin{bmatrix} c_3 & 0 & -s_3 & a_3c_3 \\ s_3 & 0 & c_3 & a_3s_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_4^3(\vartheta_4)=\begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4(\vartheta_5)=\begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_6^5(\vartheta_6)=\begin{bmatrix} c_6 & s_6 & 0 & 0 \\ s_6 & -c_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The multiplication of the single matrices to obtain the homogeneous matrix is calculated in the mathematical software Maple. With the use of Maple we obtain the C code, and introducing that set of equations into a Visual C++, the program generates a .DLL file for calculate the forward kinematics of the robot.

$$T_6^0(q) = A_1^0A_2^1A_3^2A_4^3A_5^4A_6^5$$

$$T_6^0 = \begin{bmatrix} n^0 & s^0 & a^0 & p^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$p_6^0 = \begin{bmatrix} 80((c_{123}-c_1s_{23})c_4 - s_{14})s_5 - 80(-c_1c_2s_3 - c_{13}s_2)c_5 + 290c_{12}s_3 + 290c_{13}s_2 + 75c_{123} - 75c_1s_{23} + 250c_{12} + 150c_1 \\ 80((s_1c_{23}-s_{123})c_4 + c_1s_4)s_5 - 80(-s_1c_2s_3 - s_{12}c_3)c_5 + 290s_{13}c_2 + 290s_{12}c_3 + 75s_1c_{23} - 75s_{123} + 250s_1c_2 + 150s_1 \\ 80(s_2c_3 + c_2s_3)c_4s_5 - 80(-s_2s_3 + c_{23})c_5 + 290s_{23} - 290c_{23} + 75s_2c_3 + 75c_2s_3 + 250s_2 \end{bmatrix}$$

$$n_6^0 = \begin{bmatrix} (((c_{123} - c_1 s_{23})c_4 - s_{14})c_5 + (-c_1 c_2 s_3 - c_{13} s_2)s_5)c_6 + (-(c_{123} - c_1 s_{23})s_4 - s_1 c_4)s_6 \\ (((s_1 c_{23} - s_{123})c_4 + c_1 s_4)c_5 + (-s_1 c_{23} - s_{12} c_3)s_5)c_6 + (-(s_1 c_{23} - s_{123})s_4 + c_{14})s_6 \\ ((s_2 c_3 + c_2 s_3)c_{45} + (-s_2 s_3 + c_{23})s_5)c_6 - (s_2 c_3 + c_2 s_3)s_{45} \end{bmatrix}$$

$$s_6^0 = \begin{bmatrix} (((c_{123} - c_1 s_{23})c_4 - s_{14})c_5 + (-c_1 c_{23} - c_{13} s_2)s_5)s_6 - (-(c_{123} - c_1 s_{23})s_4 - s_1 c_4)c_6 \\ (((s_1 c_{23} - s_{123})c_4 + c_1 s_4)c_5 + (-s_1 c_2 s_3 - s_{12} c_3)s_5)s_6) - (-(s_1 c_{23} - s_{123})s_4 + c_{14})c_6 \\ ((s_2 c_3 + c_2 s_3)c_{45} + (-s_2 s_3 + c_{23})s_5)s_6 + (s_2 c_3 + c_2 s_3)s_4 c_6 \end{bmatrix}$$

$$a_6^0 = \begin{bmatrix} ((c_{123} - c_1 s_{23})c_4 - s_{14})s_5 - (-c_1 c_2 s_3 - c_{13} s_2)c_5 \\ ((s_1 c_{23} - s_{123})c_4 + c_1 s_4)s_5 - (-s_1 c_2 s_3 - s_{12} c_3)c_5 \\ (s_2 c_3 + c_2 s_3)c_4 s_5 - (-s_2 s_3 + c_{23})c_5 \end{bmatrix}$$

**Figure 4.4** Equations of forward kinematics

The .DLL file for calculates the forward kinematics of the robot is introduced directly in the LabVIEW program as a call library function node to compute the Cartesian coordinate system of the LR Mate 200i. The advantage of this method is to reduce the number of variables to read from the robot controller, and obtain an increment in the performance of the data acquisition system. The example of the program is shown in the next figure.

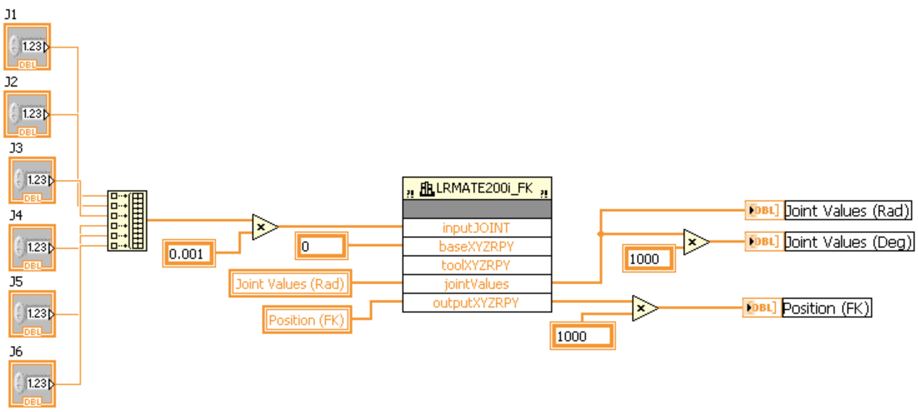



**Figure 4.5** Forward Kinematics program example

## 4.2. Accessing Torque Variables

The torque or moment of force is the tendency of a force to rotate an object about an axis. In the R-J3 controller, the system displays the disturbance torque of each motor as a current values in amperes estimated from the difference between the scheduled and actual values of the pulse coder.

The system displays the following three parameters: current torque, maximum and minimum torque for each internal cycle. If the maximum or minimum value set for the disturbance torque is exceeded, the system trigger an alarm of collision detection and turns the servo power off.

| Name | Variable | Type |
|---|---|---|
| Minimum Disturbance Torque | $MOR_GRP[1].$MIN_DIS_TRQ[9] | SHORT |
| Maximum Disturbance Torque | $MOR_GRP[1].$MAX_DIS_TRQ[9] | SHORT |
| Current Disturbance Torque | $MOR_GRP[1].$CUR_DIS_TRQ[9] | SHORT |
| Average Torque | $MOR_GRP[1].$TORQUE[9] | SHORT |
| Maximum Torque | $MOR_GRP[1].$MAX_TORQUE[9] | SHORT |
| High Precision Disturbance Torque | $MISC[1].$HPD_TRQ[9] | REAL |

**Table 4.2** Disturbance torque variables

The disturbance torque variables are shown in the table 4.2 and can be access with the method shown in the figure 3.10. The high precision disturbance torque variable shows the disturbance torque in percentage of each servo motor. This value is more precise than “$MOR_GRP.$CUR_DIS_TRQ[9]” and the system updates this variable automatically. The variable high precision disturbance torque enable “$MISC_MSTR.$HPD_ENB” must be set to TRUE in order to update the variable “$HPD_TRQ[9]”.

The current disturbance torque and the high precision disturbance torque are available only when the Soft Float option is disable. However, the calculation of the torque is possible through reading the variables “$MOR_GRP[1].$ERROR_CNT[9]” and “$MOR_GRP[1]. $MACHINE_PLS[9]”.

The variable “$ERROR_CNT[9]” returns the error, in pulse counts, from the actual position as seen by the encoders compared to the command position. The absolute encoder pulse count is read from the current robot position, and is taken from the variable “$MACHINE_PLS”. With those two variables is how the system calculates the force input.

With the position values of the robot in Cartesian coordinates and the force input in terms of percentage, the connection with the virtual model in IHA3D of the Water Hydraulic Manipulator is the next step for the implementation.

## 4.3. IHA3D Connection

The IHA3D software displays the 3D simulation of the WHMAN, where the robot orientation and movements are visualized. The communication between LabVIEW and IHA3D is over a network connection with the User Datagram Protocol (UDP). Once that the project of the WHMAN is initialized, the incoming and outgoing network connection tools must be started for establish the communication.

The IP configuration and port access could be modified, but should be identical as the IP selected in LabVIEW. The figure 4.6 shows the complete program for the configuration of the communication and the objects used for the calculation of the kinematics of the robots.

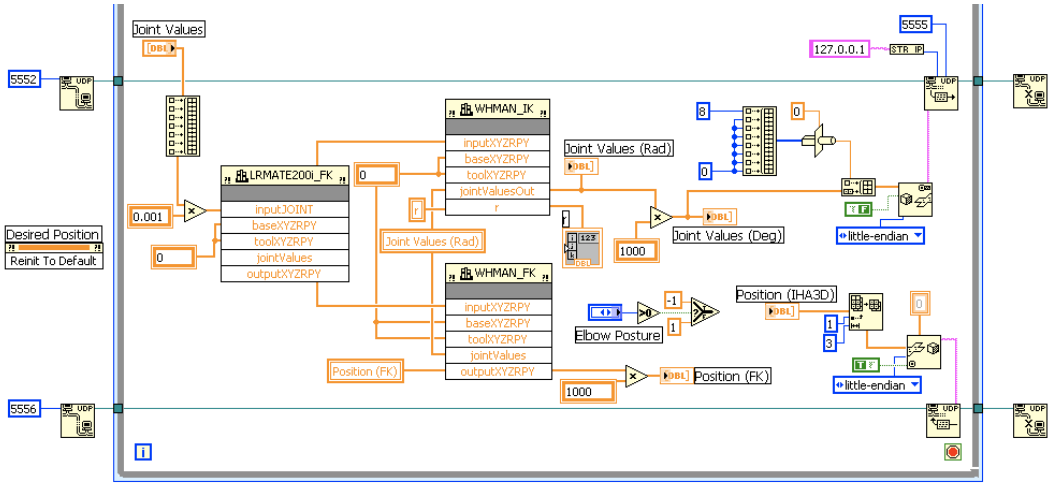



**Figure 4.6** LabVIEW program for connection with IHA3D

The IHA3D utilizes a reference frame to display the orientation of the links, and the LR Mate 200i sends the values in Cartesian coordinates. However, a compensation of the values in the coordinates system is required. The table 4.3 shows the values of the coordinate system and the compensation for the robots in a home position.

| Joint | Cartesian | Comp. | IHA3D | Cartesian | Joint |
|---|---|---|---|---|---|
| J1: 0 | X: 520 | -520 | 0 | X: 1143.496 | J1: 448.47 |
| J2: 0 | Y: 0.00 | +200 | 200 | Y: 225.5 | J2: -90.0 |
| J3: 0 | Z: 325 | +1675 | 200 | Z: -1780.889 | J3: 33.976 |
| J4: 0 | W: -180 | +180 | 0 | W: 0 | J4: 0 |
| J5: 0 | P: -90 | +90 | 0 | P: 147.296 | J5: -111.764 |
| J6: 0 | R: 0.00 | 0 | 0 | R: 0 | J6: 0 |
| | | | 0 | | J7: 77.787 |
| | | | 0 | | J8: 0 |

**Table 4.3** Values of the coordinate systems and compensation

For manipulation strategic of the master arm, the system requires the activation of digital inputs. The design of an end-effector with such capabilities is the next point.

## 4.4. End-Effector Design

The design of a mechanic interface to manipulate the robot end-effector was conceived with the next two premises: simple as its possible and with the capacity of activate digital inputs. Following the guidelines of an ergonomic design for a hand manipulation tasks, a cylinder design was selected due the human adaptability and facility to manufacture. The activation of the robot input is through a normally open pushbutton switch with a soft rubber cover, mounted at the front top of the device.

The design allows the possibility to increase the number of inputs for the manipulation and fulfill the remote handling tasks. The design was made in the CAD software CATIA and the figure 4.7 shows the 3D drawing of the mechanic interface.

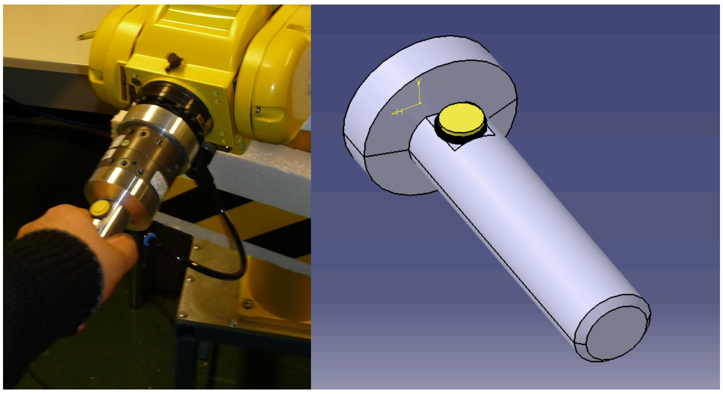


**Figure 4.7** End-Effector tool and the 3D drawing

The mechanical coupling of the handler interface and the robot end-effector is through a pneumatic robot adaptor.  The pneumatic adaptor allows the possibility of detach the device via the activation of an integral internally mounted solenoid valve pack. The electrical activation of the solenoid valves is commanded by the digital outputs of the robot.

The electrical connection for the pushbutton is through the end-effector connector built into the wrist and activated with the robot digital input signal interface. Normally the interface utilizes a connector manufactured by Hirose Electric, but the interface terminal was changed to a M12 sensor connector because is a common interface. The modification in the electrical connection and new cabling configuration is presented in the next table.

| Cable | Signal | Modification | Color |
|---|---|---|---|
| 1 | RDI1 | 1 | Brown |
| 2 | RDI2 | 2 | White |
| 3 | RDI3 | 3 | Blue |
| 4 | RDI4 | | |
| 5 | *HBK | | |
| 6 | +24v | 4 | Black |
| 7 | +24v | 5 | Gray |
| 8 | 0v | | |

**Table 4.4** End-effector connector modification

The pushbutton activates the robot input RDI1 and was connected between the cables 1 and 4. The signal displays a Boolean indicator in the LabVIEW interface that shows when the button is pressed. The description of the interface designed is detailed in the next paragraphs.

## 4.5. Monitoring Interface Developed

The interface is the core of this theses and it's where the operator visualizes the position values of the master and slave arms, and should be useful to interpret the data and have a better understanding of what happens in the remote environment.

The window interface is shown in the figure 4.8 and is divided in two parts. The first part that is in the left side of the window contains the data of the LR Mate 200i robot controller. The interface displays the position of the robot in Joint Coordinates, Cartesian Coordinates and the torque on each axis. Also provide a set of indicators to know which is the state of the digital inputs and outputs in the robot. Finally the "Connected" indicator turns on when the PC establishes the connection with the robot controller.

The second part of the interface is related to the control of the Water Hydraulic Manipulator and is located in the right side of the window. The interface displays the position of the WHMAN in Cartesian coordinates, Joint Coordinates, the position in the IHA3D model, and the force feedback values.

The force feedback is simulated in the interface and is required in this phase of the project because the teleoperation system only interacts with the virtual environment. The force feedback simulation is also supported in the IHA3D software, but for increase the system performance the simulation is realized in LabVIEW. The values of the Soft Ratio correspond to the servo flexibility bars, linked with the Cartesian Soft Float of the robot controller.

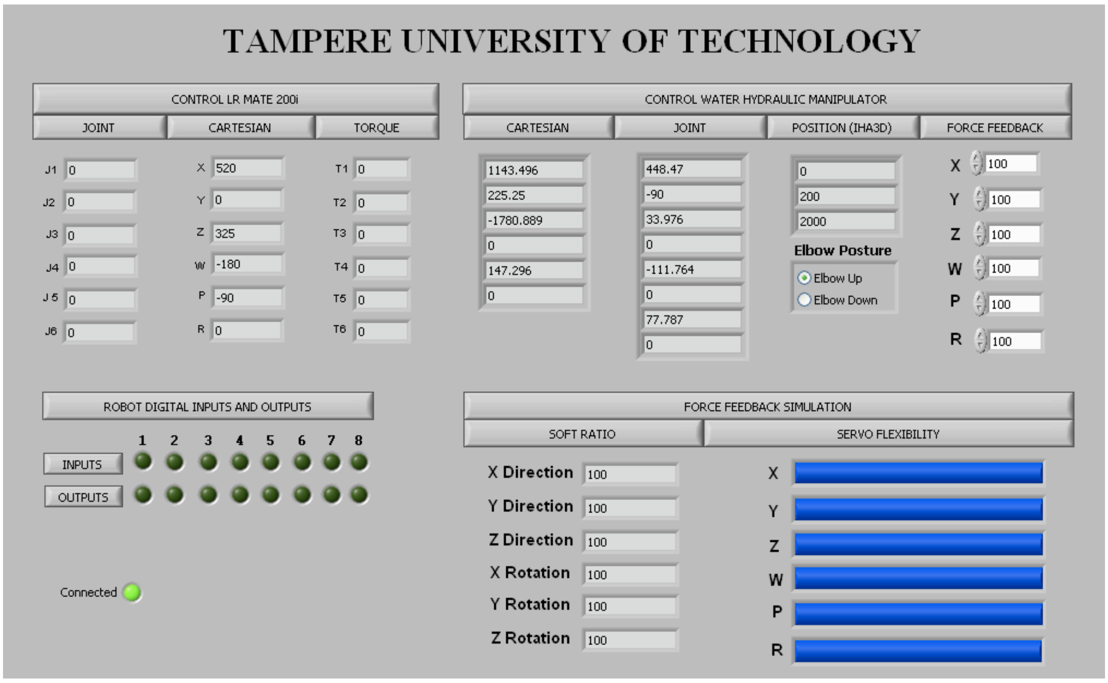



**Figure 4.8** LabVIEW interface

The configuration of the frame position must correspond in the robots. In the IHA3D software, the Tool Frame could be assigned in the Link 7 of the WHMAN model, while in the LR Mate 200i is assigned in the User Frame. For the Reference Frame, both programs utilize the nomenclature of World Frame for the coordinate position.

The difference in the structure of the robot creates the necessity of utilize the Cartesian coordinates for coupling the robot movements. The figure 4.9 shows the LR Mate 200i in home position and the WHMAN in its corresponding orientation. The orientation and frame assigned could vary depending of the required task.

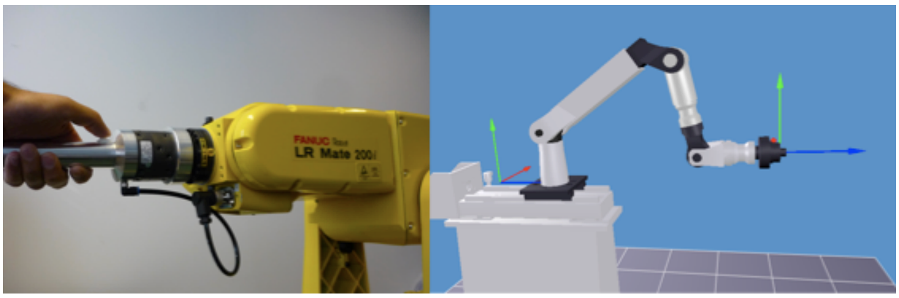



**Figure 4.9** LR Mate 200i and WHMAN in home position

# 5. RESULTS AND DISCUSSION

The execution speed in the teleoperation system is critical, due the effect of the transparency; and in order to achieve an acceptable level of transparency, the system must take into account every possible detail to increase the performance in communication, data processing, or image rendering. The results of this thesis implementation are described and discussed in this chapter with aim to be objectives.

The first part of the system to analyze is the robot controller. At variable level, the robot controller creates groups of variables to save the values and computes all the mathematical calculation required for the robot control; the number of system variables is over 7000 and vary depending on the options installed in the robot controller.

One of the most important parameter is the value of the variable "$SCR.$ITP_TIME', this variable sets the time in milliseconds of a basic motion cycle, where the minimum value is 4 milliseconds and the maximum is 124 milliseconds. If the setup of the robot controller it's at the minimum value, the system cycle is 250 Hz. Even without considering the slow access to the position and torque variables in the robot library, and its data transfer over Ethernet, the system is far away from the minimum required sampling rate of 1 kHz [19].

This limitation in the internal cycle is due the complex kinematics calculations, such as inverse Jacobian, and the different kinds of filters selected according to the configuration and the type of programs utilized for running in the RJ-3 controller. The next point to analyze is the data acquisition performance, where the connection between the robot controller and the PC is through the library of Robot Server.

The Robot Server reads the data directly from the robot, and each time that a value is written, it is immediately sent to the robot. The data access is performed through the TCP/IP network protocol, and each call requires an overhead processing time. This overhead processing time accounts for the majority of the data transfer time between the PC and the R-J3 controller.

The use of pipes for collect data increases the performance in the communication with the limitation of package size of under than 10 Kbytes, but the maximum scan rate for a stable connection with the robot controller is with a latency time of 40 milliseconds or 25 Hz.

The performance of the data processing in the LabVIEW program is another point to analyze. The program utilizes a Timed Loop to execute the communication, data display, calculate forward and inverse kinematics. This Time Loop executes the commands inside the loop at the same period as the specified in the clock.

For this implementation the loop-timing source utilizes a 1 kHz clock, with a period of 1 millisecond, as the minimum required sampling rate. The Timed Loop could be configured with a 1 MHz clock and execute an iteration every microsecond, but this timing source is only supported with the LabVIEW Real-Time Module.

The connection with the IHA3D software utilizes the UDP protocol, the constraint for incoming UDP packages is a maximum size of 960 bytes at the same sampling rate of the Time Loop in LabVIEW. For the outgoing packages the delay time for sending packages is limited to 5 milliseconds, giving a performance around 200 Hz.

The expected performance shows us that the bottleneck of the system is situated in the communication between the robot controller and the PC, with a frequency of 25 Hz. The results taken from the teleoperation system over manipulation task are shown in the next graphs.

The figure 5.1 shows the graph obtained measuring the position of the axis number 1 in Joint format. The variable taken from the Current Group Position method returns the value of the joint angle in degrees of the specified axis. The robot path was planed to move from 0 degrees to 45 degrees with a Joint motion format and CNT position path, and then move it to -45 degrees and back to 0 degrees with the same moving conditions.

The feed rate was specified at 25 %, for the axis 1 represents a speed of 4.95 degrees per seconds. The execution of the program takes 40 seconds, where the system stores 21 samples in a medium performance binary measurement file (.TDM), giving a resolution of 8.57 degrees per sample. The sampling rate is around 0.55 Hz where the expected is a maximum of 25 Hz. Is important to highlight that the results are also affected for the complexity of the method for measurement and data storage.

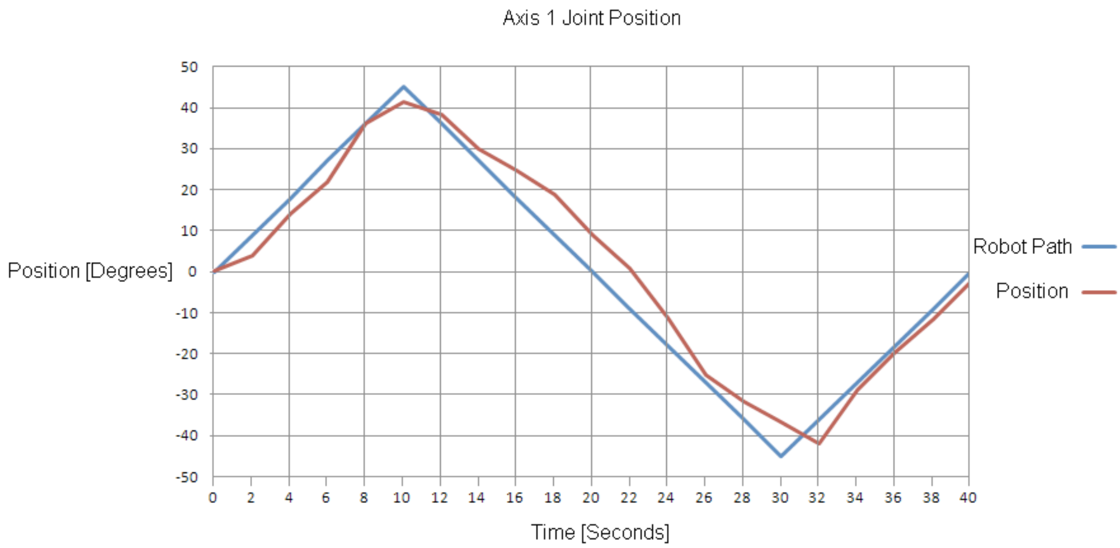



**Figure 5.1** Graph of position measurement

The resolution of the absolute encoder taken from the data sheet is 32,000 pulses per revolution or 177.77 pulses per degree. At the speed of 4.95 degrees per seconds, the encoder generates 879.96 pulses per second, this is the maximum sampling rate and compare it with the actual results is an enormous difference with the rate of 0.55 samples per second.

Another parameter that is a relevant measure is the axis torque. The figure 5.2 shows the graph of the torque measured from the axis number 5, through the variable

Current Disturbance Torque. The results were acquired with the same method conditions as the last graph and the procedure was applying a constant load of 1.78 kg at the end-effector. The distance between the applied load and the axis 5 is 23 cm.

The applied constant load generates a current of 0.8 amperes in the variable of the Current Disturbance Torque. According to the parameters of the robot controller, that are automatically updated and are preconfigured, the maximum disturbance torque for the axis 5 is 1.7 amperes, and the minimum disturbance torque is -1.7 ampere.

From the robot data sheet, the maximum value of the moment in the joint 5 is 55.5 kg*f*cm, this value correspond to 5.439 N*m. Matching the range values of amperes to the range of torque in Newton meter, the graph shows the values of the torque in the axis 5 around 4 N*m. As we can see in the graph, the measured torque takes around 2 seconds to reach the maximum value, and then oscillates around the same value.

The standard deviation of the result is 0.071, and the percentage of error is 6.25%. The accuracy of the measure is relative low as the expected, even taking in to account that the measurement was with the robot in static conditions. The assumption is that inertial effects of the robot arm are neglected due the low velocity of the operation.

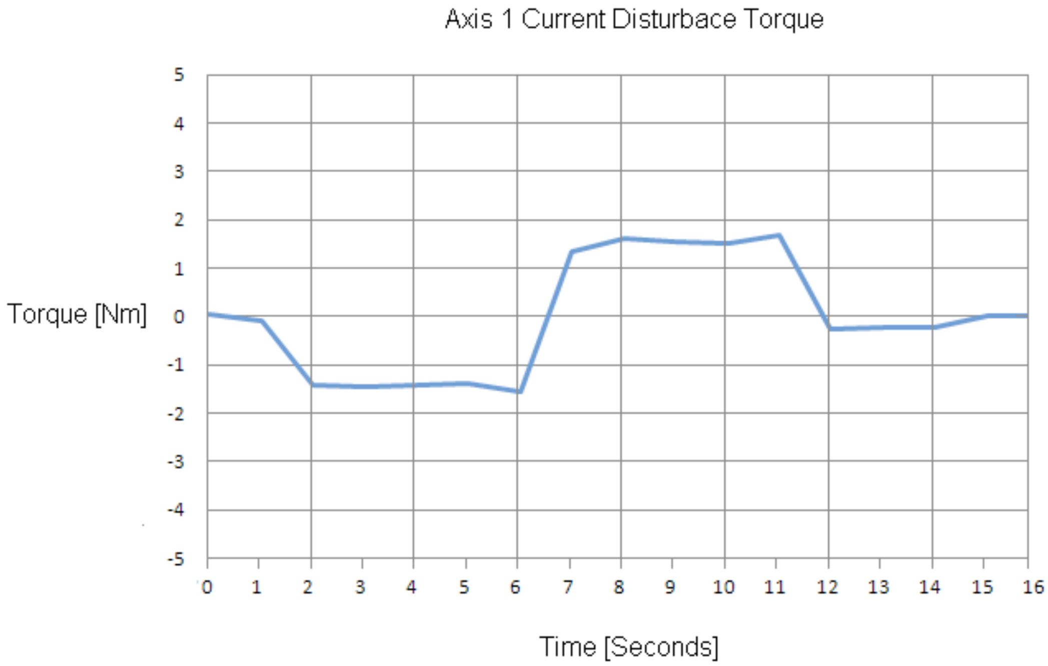



**Figure 5.2** Measured torque in Axis 1

The calculation of the torques with the robot in Soft Float mode is less accurate than the results presented in this graph. The problem relies in the lack of a stable value to get the error in the difference with the expected value. This low performance affects the control of the entire teleoperation system.

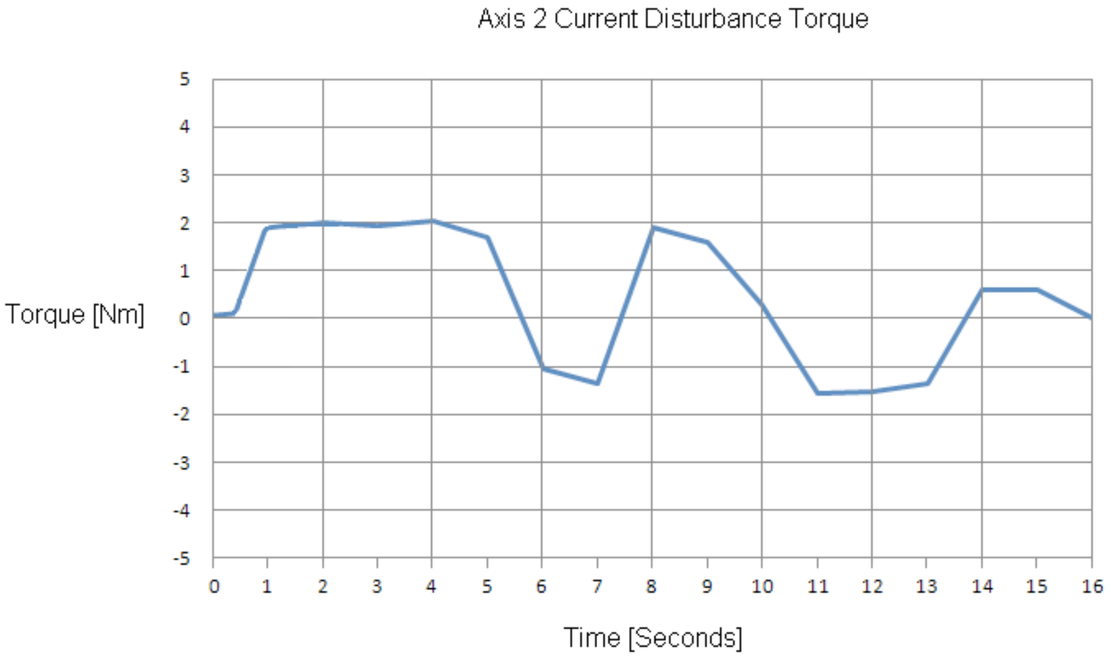



**Figure 5.3** Measured torque in Axis 2

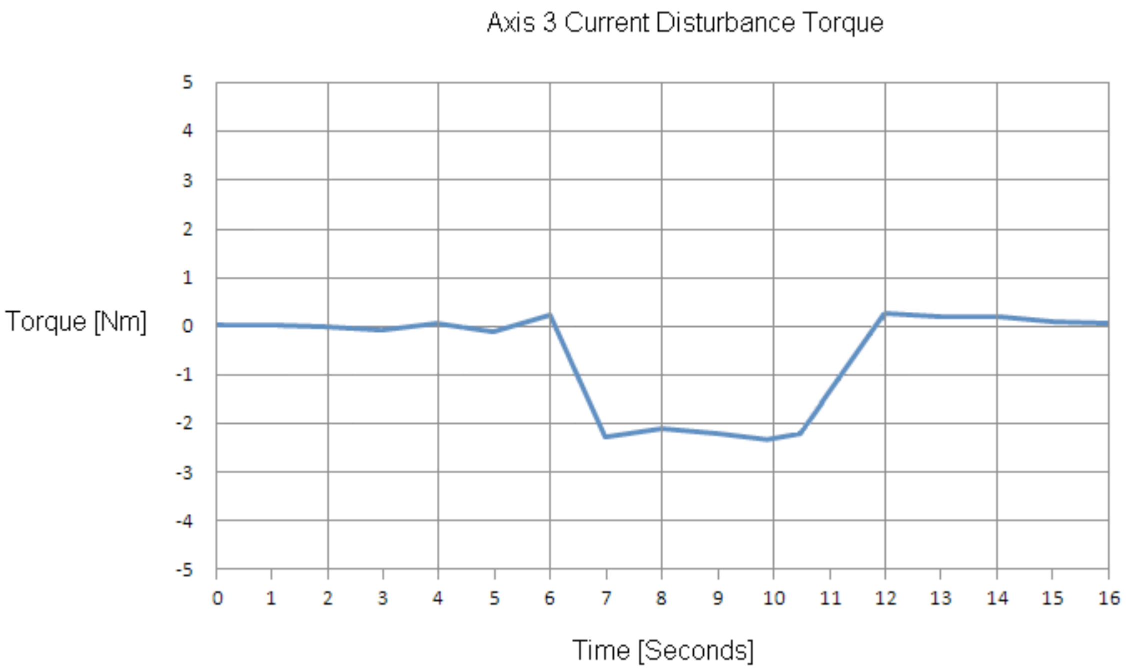



**Figure 5.4** Measured torque in Axis 3

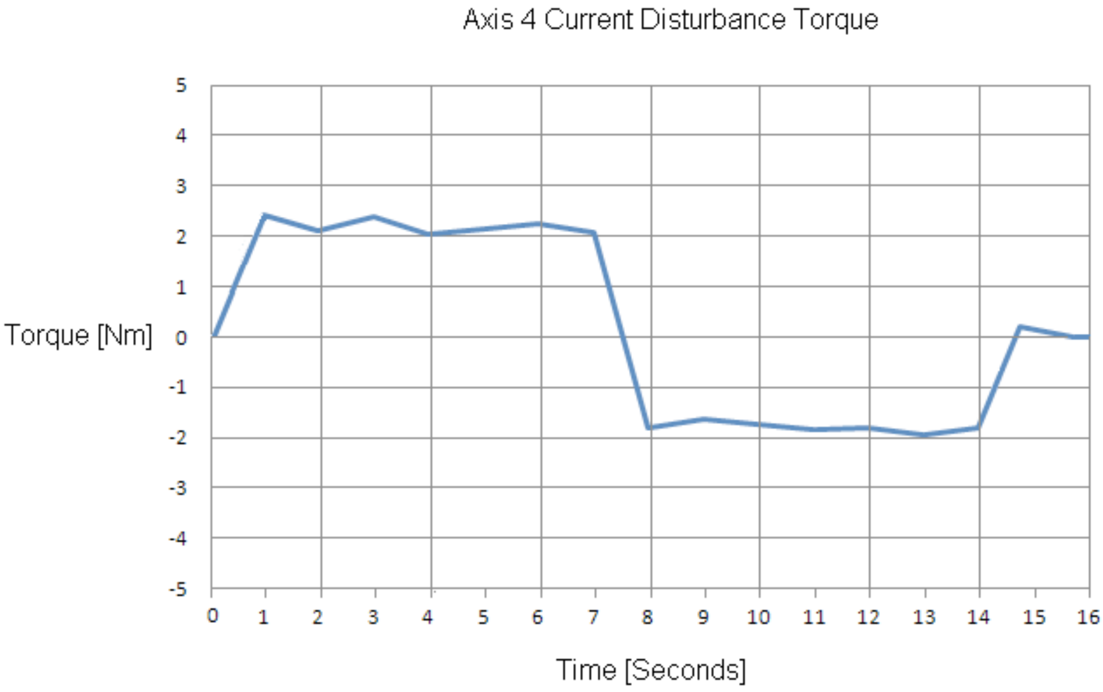



**Figure 5.5** Measured torque in Axis 4

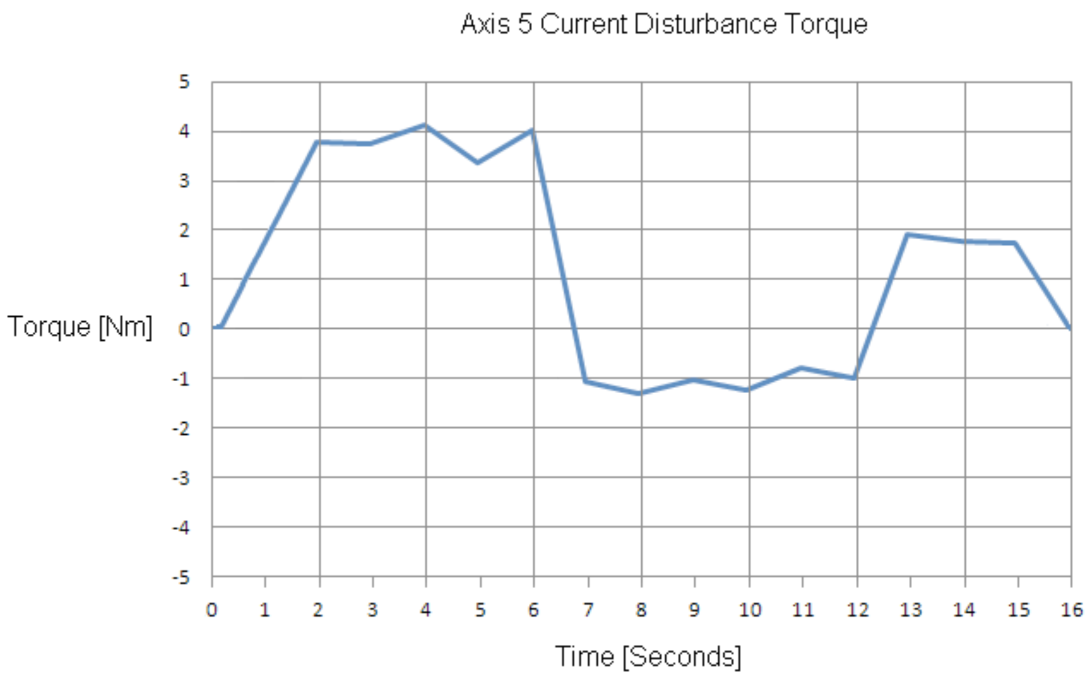



**Figure 5.6** Measured torque in Axis 5

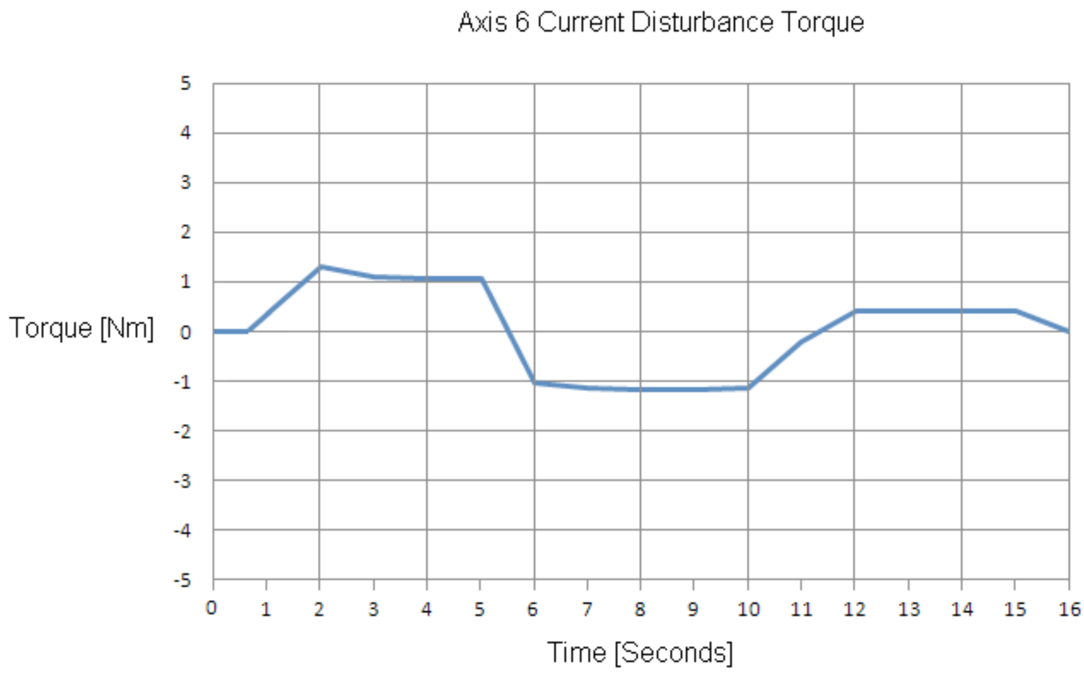



**Figure 5.7** Measured torque in Axis 6

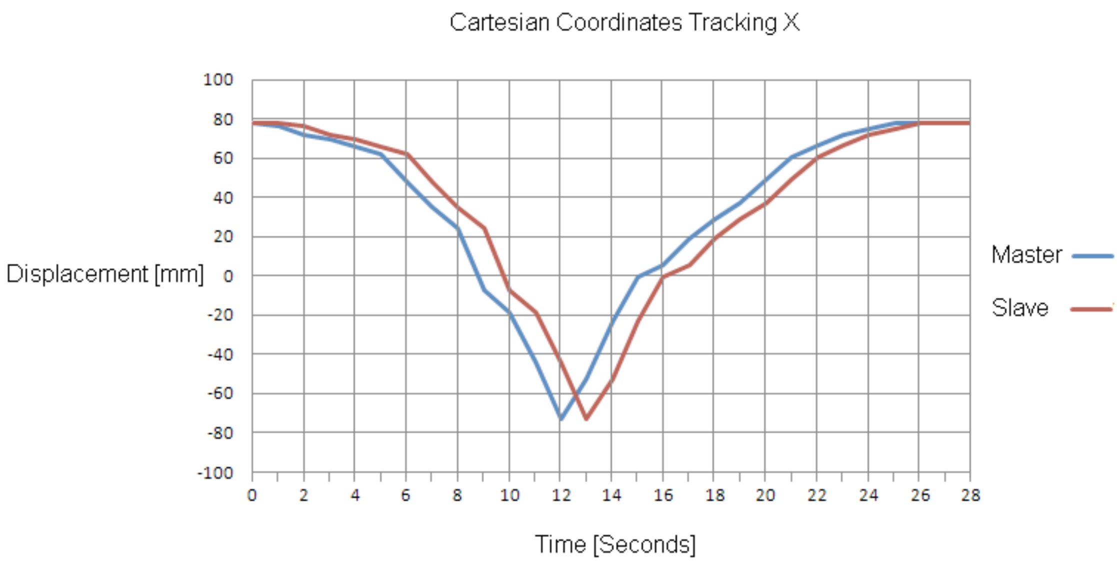



**Figure 5.8** Graph of tracking X position

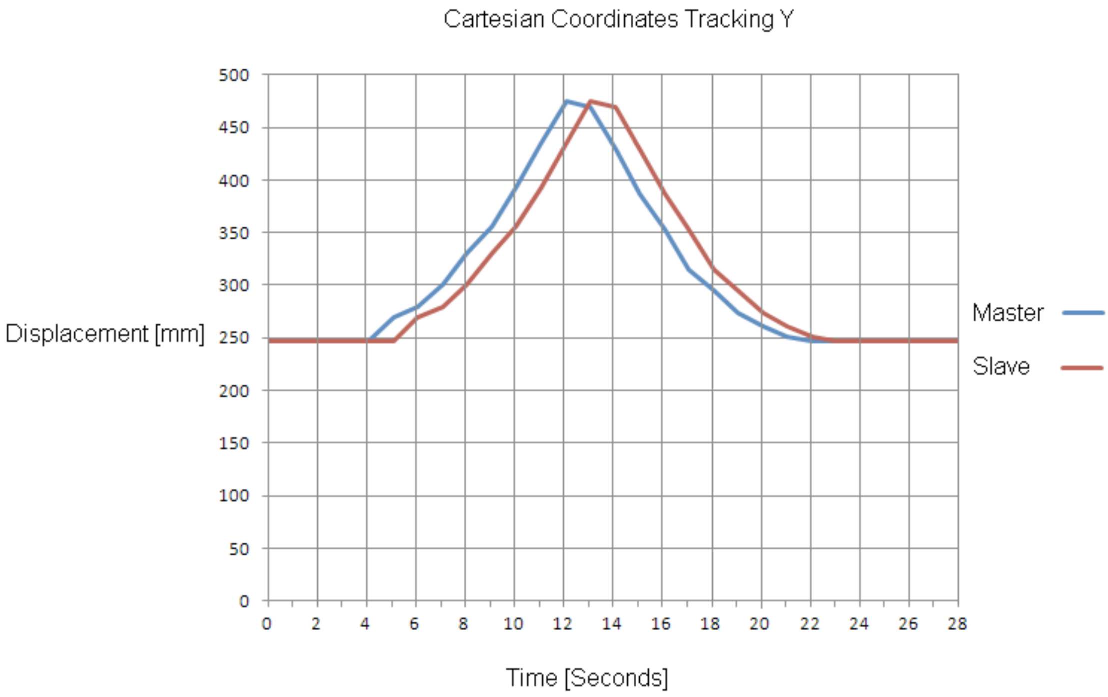



**Figure 5.9** Graph of tracking Y position

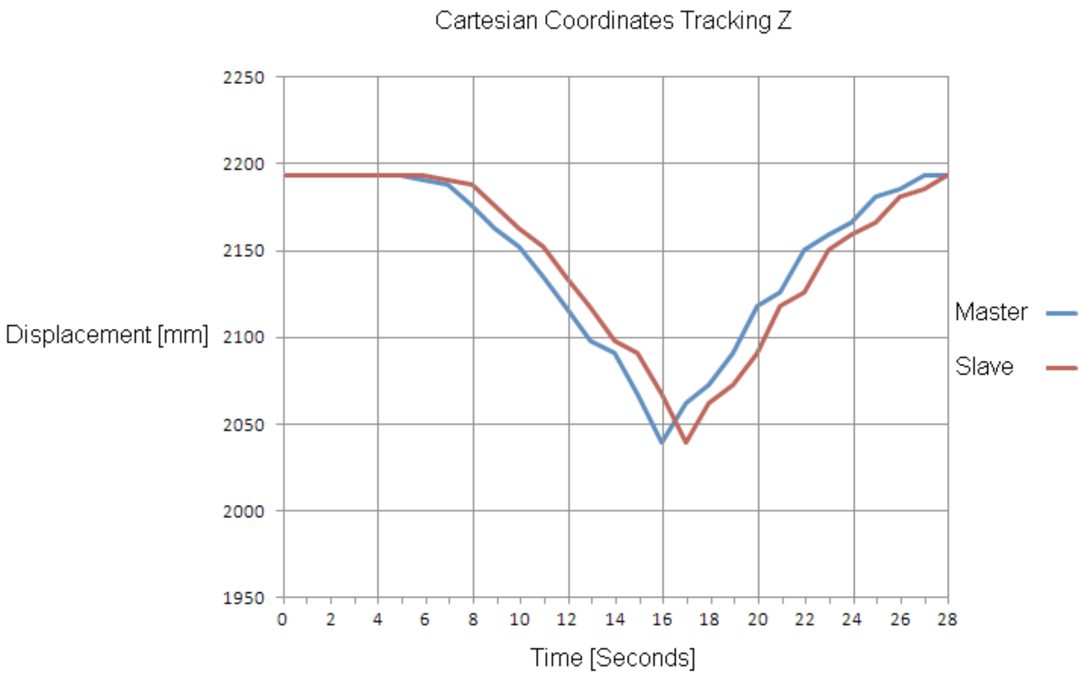



**Figure 5.10** Graph of tracking Z position

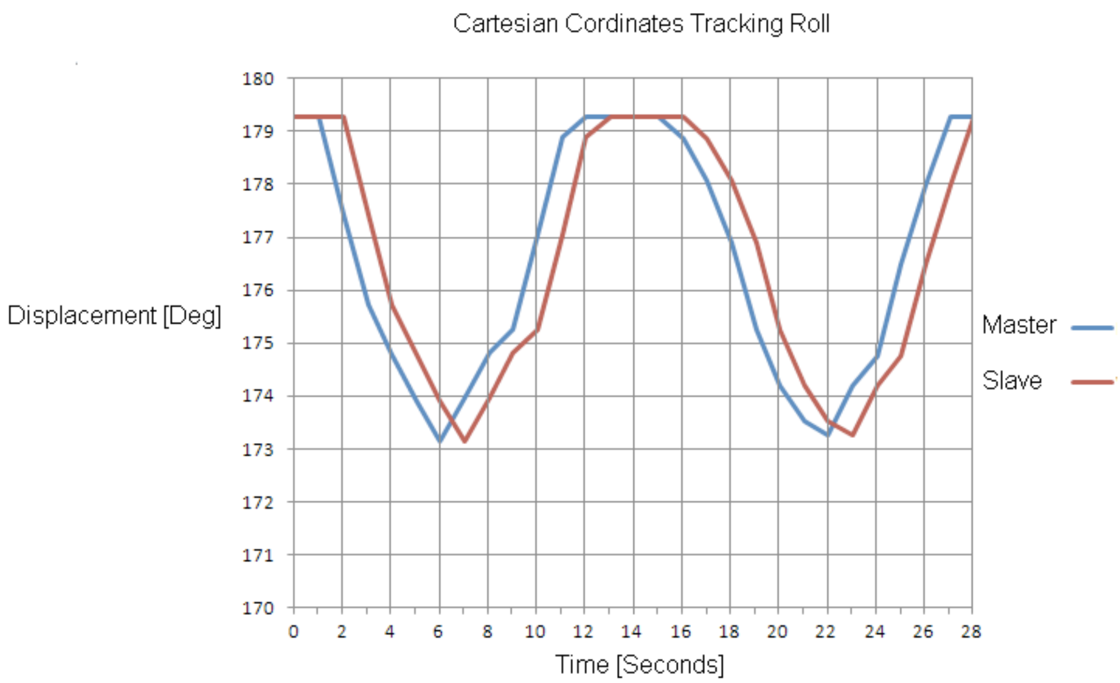



**Figure 5.11** Graph of tracking roll

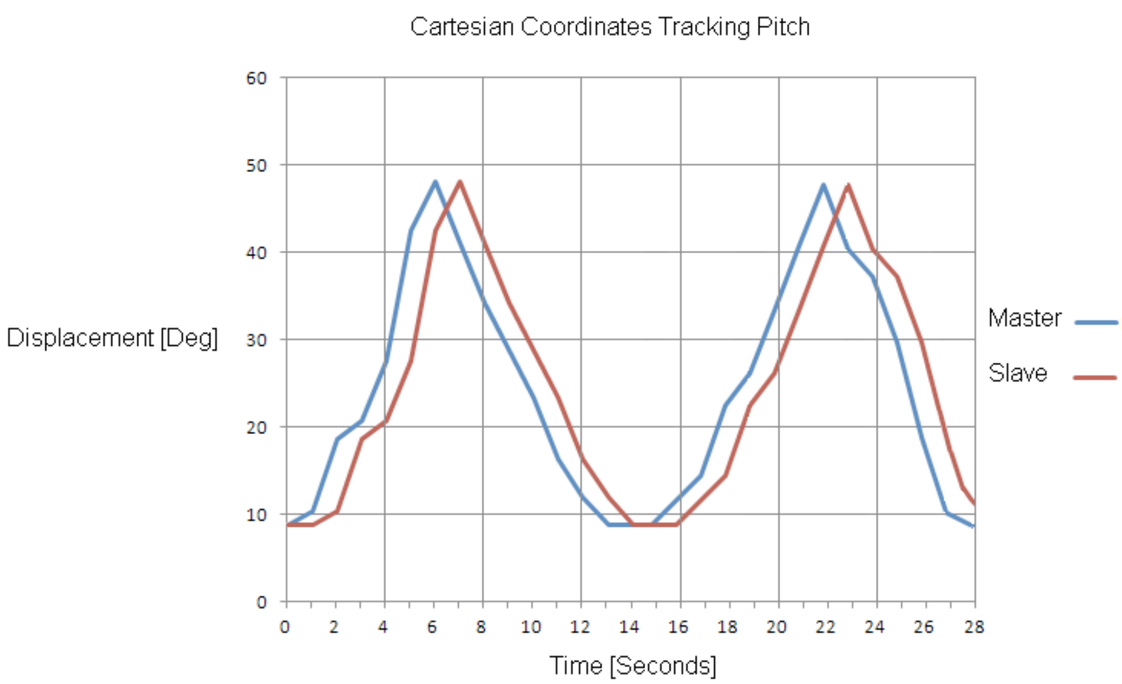



**Figure 5.12** Graph of tracking pitch

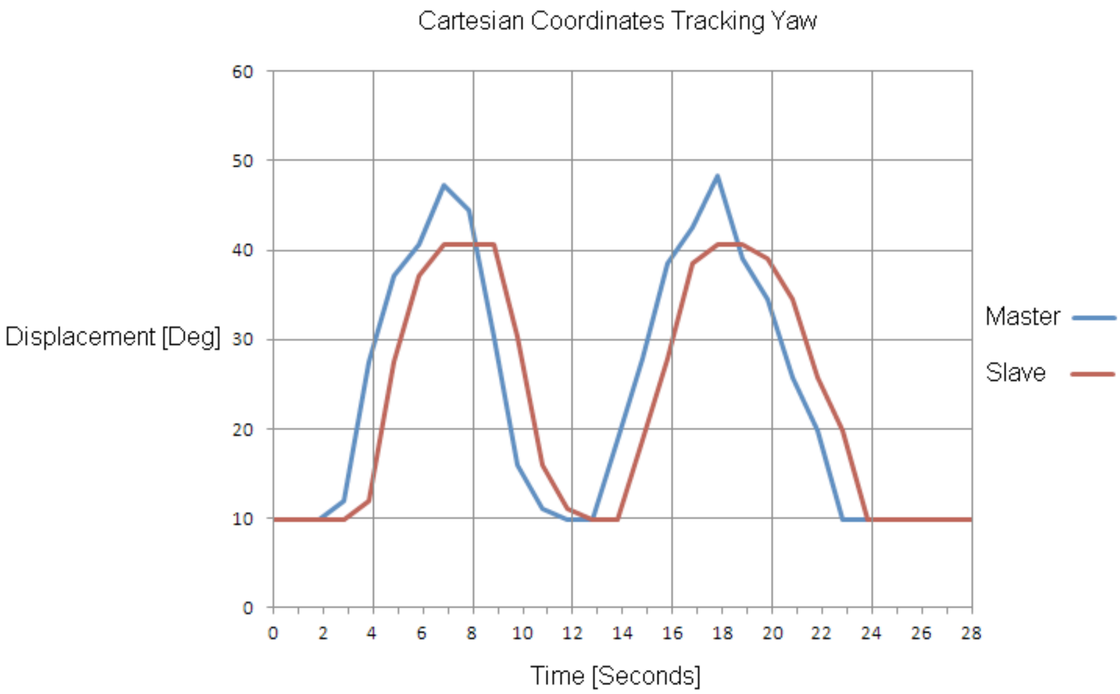



**Figure 5.13** Graph of tracking Yaw

# 6. CONCLUSIONS

The integration between the industrial robot and a third party software has been structured and documented; the user-friendly platform reduces the time of programming and facilitates the development of robotics applications.

The teleoperation system prototype was implemented, however the experimental results show an unsatisfactory level of performance. The system prototype has many unsolved problems due the short time for the implementation and requires more time for development of the robust integration.

Although the prototype is not able to fulfill the requirements of the teleoperation system for the ITER project, is a good beginning and provides an interesting tool for educational purposes in the field of robotics and teleoperation.

The research in this field has generated similar systems, trough several years of development, which proves the use of general-purpose robots as haptic interface device. This research provides a prototype that is near in performance to those systems, but with the use of less resources and time.

The basic purpose of this thesis is to probe if the industrial robot is suitable for teleoperation system; the knowledge and experience generated through the research shows that the structure of the FANUC LR Mate 200i and the R-J3 Robot Controller can be modified to create a teleoperation system with acceptable level of performance.

As elemental rule, every system has its limitation but with the enough time for development, this system has the potentiality to achieve real time performance and acceptable level of transparency. A possible problem could be related with the friction produced in the reducer unit for each axis.

Finally, the advantage of use an industrial robot for teleoperation system provides a robust device with greater range of operation in workspace and greater torque capability than the actual state of the art haptic devices in the market. And with the vision of expand the area of application the system could be a useful tool for areas like training simulators, military, research, entertainment, virtual reality environment, and not only in the nuclear industry.

## 6.1. Recommendations Future Studies

Among the main limitations of the implementation one of the most important was the performance of the system to measure the torque at the end-effector, although the system calculates the force input, the addition of a six-axis force torque sensor is recommended. The use of a transducer with a data acquisition system will provide real time measures of all six force and torque components, facilitating and improving the control of the force feedback system.

The installation of LabVIEW Real-Time Module and the components with the same capabilities is an obligatory step in future implementations. The increase of sampling rates and reduction of data processing time are the bases to develop a complete PC based robot control; which is the goal of future studies.

The control of the 6 AC servomotors in the robot by a digital servo amplifier is possible through the connection with the CPU by an optical fiber cable. The studies of the high-speed protocol FSSB (FANUC Serial Servo Buss) are required to control the servo amplifier without been limited for the software architecture and get capability to run at 50 Mhz.

The development of high performance software tools and accurate dynamics models of the industrial manipulator are essential to overcome limitations, like the friction and hysteresis effects of the reducer for each joint. The design of control schemes is another area of opportunity to compensate those effects, and the planning of control strategic for perform different task as well.

# REFERENCES


[1] The Oxford Dictionary of English. "English Dictionaries & Thesauruses". 2nd edition revised.

[2] L.Sciavicco and B. Siciliano. "Modelling and Control of Robot Manipulators". Second Edition, Springer 2005.

[3] ISO Standard 8373, Manipulating Industrial Robots – Vocabulary. 1994.

[4] E. Garcia, M. A. Jimenez, P. Gonzalez, and M. Armada. "The Evolution of Robotics Research: From Industrial Robotics to Field and Service Robotics". IEEE Robotics & Automation Magazine. March 2007.

[5] Peter F. Hokayem and Mark W, Spong. "Bilateral teleoperation: An historical survey". Coordinated Science Laboratory, University of Illinois. 2006.

[6] James P. Trevelyan, Sung-Chul Kang, William R. Hamel. "Robotics in Hazardous Applications" Springer Handbook of Robotics. 2008

[7] Sheridan T.B. and W.R. Ferrell. "Human Control of Remote Computer Manipulators". Massachusetts Institute of Technology. 1967.

[8] Thomas B. Sheridan. "Telerobotics, Automation, and Human Supervisory Control". Massachusetts Institute of Technology. 1992.

[9] J. Vertut and P. Coiffet, "Robot Technology, Teleoperation and Robotics", Vol. 3a, Prentice Hall, 1984.

[10] Antal K. Bejczy, Won S. Kim and Steven C. Venema. "The Phantom Robot: Predictive Displays for Teleoperation with Time Delay". Jet Propulsion Laboratory, California Institute of Technology. 1990.

[11] S. D. Laycock and A. M. Day. "Recent Developments and Applications of Haptic Devices". University of East Anglia. 2003.

[12] B. Hannanford and A. M. Okamura. "Haptics". Springer Handbook of Robotics. 2008.

[13] R. Bicker, D. Glennie, Ow S. M. "Telerobotics in hazardous environments". University of Newcastle upon Tyne, England. Proceedings of the international conference organized by the British Nuclear Energy Society, Leicestershire, UK, 29-30 April 1996.

[14] C.L. Clover and G.R. Luecke. "Dynamic Simulation of Virtual Mechanisms with Haptic Feedback Using Industrial Robotics Equipment". IEEE International Conference on Robotics and Automation. 1997.

[15] Y. Yokokohji, Ralph L. Hollis and Takeo Kanade. "WYSIWYF Display: A visual/Haptic Interface to Virtual Environment". Presence. Vol8, No. 4. Massachusetts Institute of Technology 1999.

[16] J. Hoogen, R, Reiner and G. Schmidt. “Control aspects of a robotic haptic interface for kinesthetic knee joint simulation”. Control Engineering Practice. 2002.

[17] M. Ponikvar, M. Munih, J. Hoogen, G. Schmidt and R. Reiner. “Haptic Environment for Analysis of Smooth Arm Movements”. Proceedin of International Conference on Advanced Robotics. 2003.

[18] Grigore C. Burdea & Philippe Coiffet, “Virtual Reality Technology”. Second Edition, John Wiley & Sons, Inc. 2003.

[19] Lawrence, D. A., “Stability and Transparency in Bilateral Teleoperation”. IEEE Trans. on Robotics and Automation, Vol. 9, No. 5, pp. 624-637, 1993.

[20] International Federation of Robotics. “2008 World Robot Market”. [Webpage] http://www.ifr.org/industrial-robots/statistics/ sited on 07-12-2009

[21] J. Martin and J. Savall. “Mechanisms for Haptic Torque Feedback”. Proceeding of the First Joint Eurohaptics Conference and Symposium of Haptic Interfaces for Virtual Environment and Teleoperator Systems. 2005.

[22] FANUC Robotic Company “Robot Model M-20iA Series” [Webpage] http://www.fanucrobotics.com

[23] Department of Intelligent Hydraulics and Automation (IHA) [Webpage] www.iha.tut.fi sited on 06-11-2009

[24] International Thermonuclear Experimental Reactor (ITER) Project [Webpage] www.iter.org sited on 13-10-2009

[25] Teemu Kekäläinen, Jouni Mattila, Tapio Virvalo. “Development and design of water hydraulic manipulator for ITER”. Tampere University of Technology, (2009).

[26] Specifications for the Phantom Premium 3.0/6DOF Haptic Device. Sensable Technologies. 1993-2006.

[27] Peetu Nieminen, Salvador Esque, Ali Muhammad, Jouni Mattila, Jukka Väyrynen, Mikko Siuko, Matti Vilenius. “Water hydraulic manipulator for fail safe and fault tolerant remote handling operations at ITER”. Tampere University of Technology, (2009).

[28] Robotics Industries Association (RIA). “Industry Statistics 2009” [Webpage] www.robotics.org sited 05-05-2009

[29] FANUC Robotics [Webpage] www.fanuc.co.jp sited 13-12-2009

[30] FRNA-10/9-DS-005. “LR Mate 200i datasheet”. FANUC Robotics North America Inc. 1999.

[31] FRNA-8/9-DS-013. “SYSTEM R-J3 Controller”. FANUC Robotics North America Inc. 1999.

[32] B-80944EN-2/01. “R-J3 Controller: Handling Tool Operator’s Manual”. FANUC Robotics. 1998.

[33] FANUC Robotics. “Software Options” FANUC Robotics (UK) LTD. 1999.

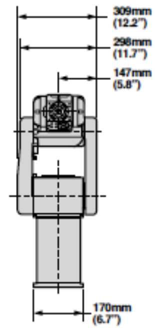
[34] Bill Moss. “Real-Time Control on Ethernet”. Dedicated Systems Magazine. 2000.

[35] FANUC Robotics. “SYSTEM R-J2 Controller: HandlingTool Setup and Operations Manual”. Version 4.40-1.FANUC Robotics North America, Inc. 1998.

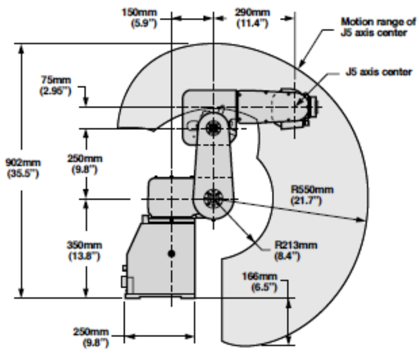
[36] John J. Craig, “Introduction to Robotics: Mechanics and Control”. Third Edition, Pearson Prentice Hall 2005.

# 10. APPENDIX 1: ROBOT AND CONTROLLER SPECIFICATIONS

**Front**




**Side**




### LR Mate 200*i* Specifications

| Items | | | |
|---|---|---|---|
| Axes | | 6 | |
| Payload | | 3kg (6.6 lbs) | |
| | | Range | Speed |
| Motion range and joint speed | J1 | 320° (± 160°) | 180°/sec |
| | J2 | 185° (+ 152°, -32°) | 180°/sec |
| | J3 | 365° (± 138°) | 225°/sec |
| | J4 | 380° (± 190°) | 400°/sec |
| | J5 | 240° (± 120°) | 330°/sec |
| | J6 | 720° (± 360°) | 480°/sec |
| Moment | J4 | 55.5 kgf • cm | |
| | J5 | 55.5 kgf • cm | |
| | J6 | 40.0 kgf • cm | |
| Inertia | J4 | 1.1 kgf • cm • $s^2$ | |
| | J5 | 1.1 kgf • cm • $s^2$ | |
| | J6 | 0.41 kgf • cm • $s^2$ | |
| Mechanical weight | | 39kg (85.8 lbs) | |
| Controller | | R-J2 Mate or R-J3 | |
| Reach | | 700mm | |
| Repeatability | | ±0.04mm (±0.002") based on JISB8432 | |
| Mounting method | | Upright/inverted | |
| Mechanical brakes | | Axis 2 and Axis 3 (Axis 1 option) | |
| Dust/water intrusion protection | | Conforms to the IP54 standard for dust and liquid protection (seals may need periodic replacement if used with chlorine or gasoline based coolants) | |

## R-J3 Standard Configuration

| Items | Specifications |
|---|---|
| *i*-size Cabinet (integrated or remote) | See drawing for dimensions |
| Operating environment | - Ambient temperature: 0-45°C<br>- Humidity: 75% RH or less non-condensing (95% max)<br>- Vibration: 0.5G or less |
| Power supply | Three phase 200-575 VAC +10%, -15%, 50/60Hz ±1Hz with circuit breaker |
| CPU | 32-bit dual processor architecture (separate motion and commun-ication) with real-time clock/ calendar |
| Controlled axes | 16 (up to three motion groups) |
| Serial/host communications | - Built-in Ethernet (10BaseT)<br>- FTP: allows simple file transfers to a variety of host platforms<br>- Ethernet Controller Backup/ Restore: provides backup and/or restore of the robot Controller memory image<br>- Three RS-232 ports (one can be configured as RS-422) |
| Teach Pendant | Back-lit LCD, multi-function |

## R-J3 Options

| Items | Specifications |
|---|---|
| B-size Cabinet | See drawing for dimensions |
| I/O sub-systems | - Model A (modular rack mounted - 5 or 10 slots)<br>- Model B (distributed DIN rail mounted) |
| I/O types | - DI/DO: 512 point maximum each (includes process I/O)<br>- Digital AC or DC input modules<br>- Digital AC or DC output modules<br>- 12-bit Analog input or output modules |
| Process I/O | - Digital input: 40 points maximum<br>- Digital output: 40 points maximum<br>- Multiple points can be utilized as a code (group I/O)<br>- Analog inputs: 6 points<br>- Analog outputs: 2 points<br>- Digital input for welding: 8 points<br>- Digital output for welding: 8 points<br>- Wire stick detect |
| Remote I/O sub-systems | - Allen Bradley Remote I/O interface<br>- Devicenet (1 channel in *i*-cabinet, up to 4 in B-cabinet)<br>- Profi-bus DP slave |
| Host communications (Ethernet-based) | - AUI (hardware interface)<br>- PC Interface: enables PC application communication |
| Diskette drive | - 3.5" HD MS-DOS format (PS--110)<br>- IBM-PC compatible disk emulator program |
| Memory card for system software installation or program backups | PCMCIA type 2 interface for:<br>- 2MB SRAM card<br>- ATA flash disk cards (SanDisk compatible) |

# APPENDIX 2: END-EFFECTOR HANDLER DRAWING